Multi-Agent Systems and Peer-to-Peer Computing: Methods, Systems, and Challenges

M. Koubarakis

Intelligent Systems Laboratory
Dept. of Electronic and Computer Engineering
Technical University of Crete
73100 Chania, Crete, Greece
www.intelligence.tuc.gr/~manolis

Abstract. In this paper we comment on the relationship between multi-agent systems and peer-to-peer systems. We point out that work in the frontiers of these two areas can be beneficial to both P2P computing and multi-agent systems, and survey recent research that is already addressing these issues. Then we summarize the work of our group at the Technical University of Crete on the topic of selective information dissemination using P2P networks of middle-agents.

1 Introduction

In P2P systems a very large number of autonomous computing nodes (the *peers*) pool together their resources and rely on each other for *data* and *services*. Popular systems such as Napster[1] (now in a transition state), Gnutella[2], Freenet[3], Kazaa[4], Morpheus[5] and others have made this model of interaction popular. Ideas from P2P computing can also be applied to other areas beyond data sharing such as Grid computation (e.g., SETIHome[6] or DataSynapse[7]), collaboration networks (e.g., Groove[8]) and even new ways to design Internet infrastructure that supports sophisticated patterns of communication and mobility [54].

The wealth of business opportunities promised by P2P networks has generated much industrial interest, and has resulted in the creation of various re-

[1] `http://www.napster.com`
[2] `http://gnutella.wego.com`
[3] `http://freenet.sourceforge.net`
[4] `http://www.kazaa.com`
[5] `http://www.musiccity.com`
[6] `http://www.setiathome.ssl.berkeley.edu`
[7] `http://www.datasynapse.com`
[8] `http://www.groove.net`

search and industrial projects[9], startup companies, and special interest groups.[10] Researchers from distributed computing, networks, multi-agent systems and databases have also become excited with the P2P vision, and papers tackling open problems in this area have started appearing in high-quality venues (such as ICDCS, SIGCOMM, INFOCOM, CIDR, SIGMOD, VLDB, etc.) but also new specialized conferences and workshops.[11]

The multi-agent systems (MAS) community has reacted rather slowly to the P2P wave as pointed out in the call for papers for the First International Workshop on Agents and Peer-to-Peer Computing (AAMAS-2002, Bologna, Italy). This reality is in direct contrast with other disciplines such as distributed systems and networks that have jumped on the P2P bandwagon and have led the development of the second generation of P2P networks based on distributed hash tables (DHT). It is also disappointing given that deployed P2P systems can be considered an interesting case of MAS as pointed out originally in [25].

In the first part of this paper we point out that the basic concepts underlying P2P systems and MAS are very similar and make a case for a *synthesis* of research agendas that would benefit both areas. Given the current state of the art, MAS can readily offer concepts and techniques that can be useful to P2P computing at the *application modeling* and *design level* (e.g., ontologies for describing network resources in a semantically meaningful way, protocols for meaning negotiation, P2P system modeling and design methodologies etc.). On the other hand, P2P computing can offer readily *popular application domains*, *sophisticated implementation techniques* and *prototype infrastructure components* for implementing and deploying MAS. As more intense cross-fertilization among these two areas takes place, we expect to see researchers from MAS and P2P crossing into each other's territory with their contributions (e.g., models of trust developed in the area of P2P systems could find applications in MAS or resource allocation algorithms pioneered in MAS can find applications to P2P computing etc.).

In the second part of the paper we summarize some of our work in the area of MAS and P2P carried out at the Intelligent Systems Laboratory of the Technical University of Crete since early 2001. The results presented emphasize the use of ideas from P2P systems to the implementation of middle-agent networks for *selective dissemination of information (SDI)*. For more details the interested

---

[9] See the European projects DIET (`http://www.dfki.de/diet`), BISON (`http://www.cs.unibo.it/bison/`)), MMAPPS (`http://www.mmapps.org`), SWAP (`http://swap.semanticweb.org/`) and the recent US project IRIS (`http://iris.lcs.mit.edu/`).

[10] For example, see the activities of the P2P working group of the Global Grid forum at `http://www.gridforum.org`.

[11] For example, see the 1st and 2nd IEEE International Conference on P2P Computing (`http://www.ida.liu.se/conferences/p2p/p2p2002`), the International Workshop on Agents and P2P Computing (`http://p2p.ingce.unibo.it`), the 1st and 2nd International Workshop on Peer-to-Peer Systems (`http://iptps03.cs.berkeley.edu`) and so on.

reader should consult the papers [37, 41, 38, 39] and technical reports [60, 33] (all of them are available on our Web site).

The rest of the paper is organized as follows. Section 2 surveys recent work at the frontiers of MAS and P2P systems and makes a case for a synthesis of ideas from these two areas. Section 3 presents the P2P middle-agent architecture we have used in our work on SDI. Then Section 4 summarizes our contributions in this area. Finally, Section 5 presents our conclusions.

2 P2P Computing and MAS: A Synthesis

Since their inception, MAS have always been thought of as *networks of equal peers.* Today, the concept of agent in MAS offers the right kind of abstraction for *enriching* the concept of peer as implemented in today's P2P systems. MAS is an appropriate framework for realizing all popular P2P applications mentioned in the introduction of this paper and many more. The typical properties associated with *agency* (e.g, *autonomy, social ability, reactivity* and *pro-activeness* [65]) are some of the characteristics that one would wish to have in the peers of next generation P2P systems. Other *society-level* properties such as *emergence of complex behaviours* and *self-organization* typically stressed in *bottom-up* approaches to MAS [23] are also very important in this context because these are exactly the kind of properties exhibited by deployed P2P networks [2, 52, 34, 4].

In the rest of this section, we survey previous work that synthesizes concepts from MAS and P2P computing and demonstrates that results from each area can be beneficially used in the other. Our presentation is selective and we concentrate on research that is closest to our work presented in Section 3 and 4.[12] We also point out some open research questions.

2.1 P2P for MAS

Existing P2P system technology can be utilized in the development of MAS in many ways. First, P2P networks can provide *infrastructure* for deploying MAS. For example, P2P networks can be used to provide Internet-wide discovery services as originally proposed for the agent-to-agent (A2A) framework of RETSINA where agents participate as nodes in the existing public Gnutella network to increase their reachability of other agent-provided services (e.g., agent name servers or middle-agents) [43]. The basic idea of [43] is in the right direction, but today there are even better choices for the P2P networks to be used. The Gnutella network is known to have scalability problems because of its use of flooding for broadcasting messages. Thus if one wishes to deploy MAS on a publicly available P2P network, one could use the more recent super-peer based P2P

[12] For example, we do not deal at all with the problem of resource allocation in P2P systems and do not discuss what MAS research has to offer to this problem. We also do not discuss other interesting problems at the frontiers of these two areas such as the existence of *non-cooperative, selfish* and *malicious* agents (peers) and the associated problems of *trust, reputation* and *security.*

networks (e.g., Morpheus[13] – currently the most popular P2P network according to [67]).

An even better alternative would be to use one of the DHT-based second generation P2P networks such as Chord that provide much better *scalability, self-organization, availability, fault-tolerance* and *load balancing* [55]. DHT-based systems could easily provide Internet-wide agent name services for MAS because all that is needed in this case is a facility for mapping agents to their IP addresses given some identifying agent feature usually a *name* [57].

Other kinds of MAS infrastructure such as middle-agent networks can also be implemented using techniques from P2P networks. While middle-agents can be classified in various different dimensions [63, 40], their basic functionality is to match *requests* for agents with a certain capability with *advertisements* specifying available capabilities. As an example, [39, 41, 38] (and Sections 3 and 4) present the theoretical principles and implementations techniques of DIAS, a distributed SDI system for digital libraries based on a P2P network of middle-agents. The middle-agent network of DIAS can be understood to follow the *content-based routing* protocol of [42], but it has been implemented using the routing algorithms of the P2P pub/sub system SIENA [13].

It is currently an open question how to use a DHT to implement more sophisticated kinds of MAS infrastructure e.g., the middle-agents of RETSINA [57] that are using service description languages such as LARKS [58] or DAML-S [49].

2.2 MAS for P2P

It is instructive to think of P2P systems in terms of MAS concepts both at the *modeling* and *design* level and at the *system architecture* and *implementation* level.

At the modeling and design level, the concept of agent and its associated properties (autonomy, social ability, reactivity, pro-activeness) can be understood as an extension of the concept of peer in today's P2P systems. Also, as shown by recent measurement studies, deployed P2P networks are self-organizing societies of agents (peers) with various emergent properties [2, 52, 34, 4]. These observations have given rise to various proposals for implementing P2P systems using MAS techniques as we will see below. What is a largely open question in this area, is how to extend current agent-based software engineering methodologies [64, 14] to the case of P2P systems. Some preliminary steps towards this area are taken in [8] where it is shown how to apply the agent-oriented software engineering methodology Tropos [14] to the development of distributed knowledge management applications on current P2P platforms such as JXTA.

At the system architecture and implementation level, Tim Finin and Yannis Labrou have suggested to think of Napster servers as *matchmakers*, a special kind of middle-agents operating using appropriate KQML protocols and a simple ontology for MP3 files [25]. Today, the use of schemas or ontologies for describing network resources will allow P2P systems to cross into the world of *semantics*

[13] `http://www.musiccity.com`

and take advantage of recently developed Semantic Web technologies.[14] This is in fact the main goal of project EDUTELLA [47] where RDFS schemata and RDF metadata can be used to annotate P2P resources, and query languages such as RDF-QEL [47] or RQL [35] can be used for search. EDUTELLA is built on top of Sun's P2P platform JXTA[15] and its current application domain is the exchange of learning materials.

In the context of the Semantic Web, *globally shared* schemas or ontologies are used to annotate Web content so that agents can operate on it in more effective ways. In the context of P2P systems, this notion of global commonly agreed ontologies can be relaxed in favour of *local* ontologies and local translations among ontologies of neighbouring peers [1, 36, 31, 30]. Through appropriate P2P communication, local agreements on semantics can give rise to *global agreement* in a *bottom-up emergent* manner [1]. It is an open question what concepts, algorithms and methodologies are appropriate for making progress in this problem.

Implementation techniques from the area of agent systems can also be used beneficially for the development of a new generation of P2P systems. BestPeer is a prototype P2P system implemented at the National University of Singapore that uses *mobile agents* as its implementation technology [48]. Routing in BestPeer is performed using flooding and TTLs like in Gnutella. However, BestPeer uses *mobile code* as its communication unit and achieves a number of advantages against first generation P2P systems such as Gnutella. First, BestPeer can achieve a finer granularity of data sharing. Local nodes can "mask" their content using *active objects* and can provide different replies to identical requests depending on the identity or privileges of the requester. Furthermore, requesters can ship code that contains *filtering subroutines* to another peer's site to select the content that will be returned based on the requester's own criteria. This feature is especially useful when the language for requests supported by the P2P system does not allow requesters to express their preferences in an effective way. By moving filtering to the provider's site, BestPeer increases bandwidth utilization because useless data is never returned to the requester. Additionally, BestPeer uses a distributed network of servers to allow *location independent global name lookup*. Using these servers, peers can maintain their unique identity even after disconnecting from the network and reconnecting from another IP address. Finally, BestPeer allows a peer to *reconfigure* its direct connections in the network to *get closer* to preferred peers (where preference can be based on criteria such maximum number or quality of answers). [48] demonstrates experimentally that this self-configuration capability allows BestPeer to outperform Gnutella in query answering performance. Leaving aside the usual permission/security considerations raised for every mobile agent system, BestPeer offers a nice agent-based framework for the development of P2P applications.

Shehory [53] presents a theoretical analysis of the practical problem of finding the location of an agent with certain capabilities in an open MAS that is like the Gnutella P2P network (i.e., there are no middle-agents so each agent keeps

[14] `http://www.w3.org/2001/sw/`
[15] `http://www.jxta.org`

a local list of other agents and their capabilities). [53] considers situations where agents are connected in a lattice-like rectangular graph structure, and shows that dramatic improvements in the number of communication operations is possible by randomly connecting a small number of pairs of nodes in the original graph. Additionally, Shehory gives some evidence that these results apply to larger classes of graphs as well. The practical relevance of this work is that it shows that if there is enough structure in the topology of the MAS then approaches with no middle-agents may result in very small communication costs.

Bottom-up Approaches to MAS An important subarea of MAS research that is expected to play an important role in the development of future P2P systems is the bottom-up approach [23] and the closely related area of *complex adaptive systems* [61, 26]. Bottom-up approaches to MAS emphasize that it is rather impossible to understand many application domains by traditional *top-down decomposition:* breaking them into components, understanding what each component does and deducing the behaviour of the whole system from the behaviour of its components. In such societies of agents the whole is greater than the sum of the parts, and the behaviour of the whole can only be understood by understanding the *interactions* between the autonomous agents that make up the whole and lead to *emergent global behaviours.*

European projects DIET[16] [44, 32] and BISON[17] [6] were the first to point out that P2P networks are clearly an instance of CAS and follow a *nature-inspired* computing approach [9, 26] to their development.

DIET started in July 2002 and one of its first results were the implementation of a MAS platform inspired by *natural ecosystems* and bottom-up approaches to Artificial Intelligence [44, 32]. In the context of DIET, an ecosystem is composed of one or more communities of living organisms interacting with each other and with the physical environment that they inhabit. At the implementation level, the DIET platform offers a minimal *core layer* that allows the development of *lightweight mobile agents* (the living organisms) occupying *environments* that are parts of *worlds* (there is one world for each Java Virtual Machine on a user's computer). Similarly with natural ecosystems, DIET agents may be very simple but their collective behaviours and the overall functionality arising from their interactions exceed the capacities of any individual agent. DIET agents can *migrate* to other environments in the same or a remote machine, and can communicate with each other using *local* or *remote* communication mechanisms. Additional application functionality can be implemented on top of the core layer. The DIET platform has been shown to scale to large numbers of lightweight agents [32] and to form a nice basis for developing various efficient, robust and adaptive applications based on the P2P paradigm: information retrieval [28], information dissemination [38, 41], content-sharing [59], community self-organization and group formation [32, 62] and distributed look-up infrastructures [10].

[16] `http://www.dfki.de/diet`

[17] `http://www.cs.unibo.it/bison/`

BISON is a European project that has commenced in January 2003. Its partners have already designed and implemented a mobile agent system called Anthill [6] which can be used to construct P2P applications based on the *swarm-intelligence* paradigm of Dorigo and colleagues [9]. The Anthill model is based on the concepts of nest and ant. A *nest* is a middleware component written in Java that can host resources and can perform computations. A P2P application built on Anthill is organized as an overlay network of interconnected nests. Nests can interact with local instances of one or more *applications* and provide them with various *services* by generating *ants* that travel the overlay network to perform the required task. Anthill is built on top of JXTA and currently includes a *run-time environment* for developing working systems and a *simulation* environment for experimenting with P2P algorithms before deploying them. Ant implementations can be the same in the run-time and simulation environments to avoid duplicate effort in developing applications.

Up to this point, Anthill has been used to develop a file-sharing application called Gnutant, and a load-balancing application called Messor [6, 45, 46]. Gnutant is similar to Freenet in functionality and is described in more detail in [6]. Messor provides an ant algorithm for balancing the load in a simple Grid computing system aimed for highly-parallel, time intensive computations in which the workload can be decomposed into many independent jobs [45, 46]. The main idea behind Messor is to have ants travelling a network of nests which host computational resources, trying to "disperse" jobs in order to balance the load throughout the network. Essentially, Messor carries out the opposite function to the one performed by several species of real ants that are known to collectively gather objects in their environments to form piles [51].

DIET and Anthill have shown the way, but there is still a lot of work to be done in this area. The first interesting problem is to compare the approaches taken by DIET and Anthill with standard ways of developing P2P applications and to evaluate the benefits and deficiencies of each approach. Furthermore, one should work towards the development of a methodology for the application of nature-inspired techniques to P2P networks. This will allow us to move away from the current *harvesting* approach where we pick an interesting heuristic from nature (e.g., the behaviour of ant colonies mentioned above) and apply it to a P2P problem (e.g., load-balancing), to a *synthesis* phase where we have a methodology for guiding the search for CAS that lead to the desired global behaviour. This is one of the goals of the BISON project as presented by Ozalp Babaoglu in [5]. Finally, let us point out that the results of this research will strongly support the "shift in paradigm" for software engineering research envisioned in [68] where it is conjectured that we are witnessing a move from the current "design-based mechanical" foundation of software engineering to a more "physics-based" or "intentional" foundation. They will also lead to the development of methodologies for the design and implementation of P2P systems, an open problem we have mentioned at the beginning of this section.

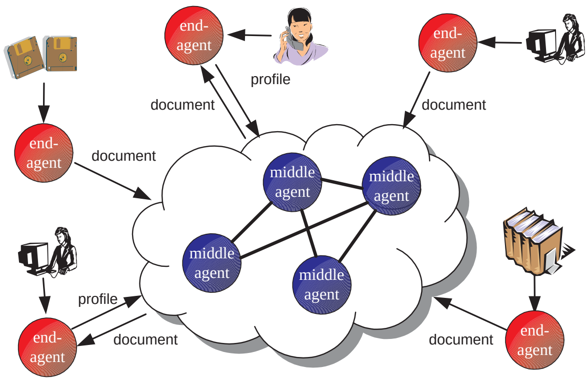

Fig. 1. A P2P agent architecture for SDI

Here we finish our survey of recent research on the frontiers of MAS and P2P. It is our belief that a synthesis of the research agendas of these two areas will be very beneficial to both.

3 P2P Middle-Agent Networks for SDI

Let us now turn to summarizing some of our recent results on MAS and P2P systems. The work presented here starts from the observation that the main application scenario of recent P2P data sharing systems is that of *ad-hoc querying*: a user poses a query (e.g., "I want MPEGs with video-clips of Jennifer Lopez") and the system returns a list of pointers to matching files owned by various peers in the network. Then, the user can go ahead and download files of interest. The complementary scenario of *selective information dissemination (SDI)* or *selective information push* [27] has so far been considered by very few P2P systems [38, 50, 29]. In an SDI scenario, a user posts a *profile* or *continuous query* to the system to receive notifications whenever certain *events* of interest take place (e.g., when a video-clip of Jennifer Lopez becomes available). SDI can be as useful as ad-hoc querying in many target applications of P2P networks ranging from file sharing, to more advanced applications such as alert systems for digital libraries, e-commerce networks etc.

We envision an information dissemination scenario in the context of a *distributed peer-to-peer (P2P) agent architecture* like the one shown in Figure 1. This architecture was originally proposed in [12] in the context of the P2P *event notification system* SIENA.[18] This agent architecture has been adopted by our group since early 2001 and it was first presented in [38].

In the scenario of Figure 1 users utilize their *end-agents* to post *profiles* or *documents* (expressed in some appropriate language) to some *middle-agents*.

[18] SIENA uses the term *server* instead of middle-agent but the idea is exactly the same.

End-agents play a dual role: they can be information producers and information consumers at the same time. The P2P network of middle-agents is the "glue" that makes sure that published documents arrive at interested subscribers. To achieve this, middle-agents forward posted profiles to other middle-agents using an appropriate P2P protocol. In this way, matching of a profile with a document can take place at a middle-agent that is as close as possible to the origin of the incoming document. Profile forwarding can be done in a sophisticated way to minimize network traffic e.g., no profiles that are less general than one that has already been processed are actually forwarded.

In their capacity as information producers, end-agents can also post *advertisements* that describe in a "concise" way the documents that will be produced by them. These advertisements can also be forwarded in the P2P network of middle-agents to *block* the forwarding of *irrelevant* profiles towards a source. Advertisement forwarding can also be done in a sophisticated way using ideas similar to the ones for profile forwarding.

Most of the concepts of the architecture sketched above are explicit (or sometimes implicit) in the KQML literature and subsequent MAS based on it [24, 42, 22, 56, 58]. Unfortunately the emphasis in most of these systems is on a single central middle-agent, making the issues that would arise in a distributed setting difficult to appreciate. In our opinion, the best presentation of these concepts available in the literature can be found in [12] and our paper [38].

4 Summary of Our Results

Our work on P2P middle-agent networks for SDI has been driven by the following considerations:

– The next generation of P2P data sharing systems should be developed in a *principled* and *formal* way and classical results from logic and theoretical computer science should be applied. Most of the current work on P2P systems has not emphasized such theoretical considerations at all, as witnessed by the recent survey of [21].
– Performance and scalability must be a primary consideration in the design of any realistic system that supports SDI with P2P networks.
– Implementations of P2P systems should be based on modern technologies that would enable rapid application development and reuse.

In the rest of this section we discuss our work up to now and what we have done to address these considerations.

4.1 Data Models and Query Languages

We have developed the data models $\mathcal{WP}, \mathcal{AWP}$ and $\mathcal{AWPS}$, and their corresponding languages for specifying queries and notifications. $\mathcal{WP}$ is based on free text and its query language is based on the *boolean model with proximity operators*. The concepts of $\mathcal{WP}$ extend the traditional concept of proximity in IR [7, 16, 17] in a significant way, and utilize it in a query language targeted at SDI for distributed digital libraries. Data model $\mathcal{AWP}$ is based on *attributes* with finite-length strings as values. Its query language is an extension of the query language of data model $\mathcal{WP}$. Finally, the model $\mathcal{AWPS}$ extends $\mathcal{AWP}$ by introducing a "similarity" operator based on the IR vector space model [7].

Our data modelling work complements recent proposals for querying textual information in SDI systems [12, 11] by using linguistically motivated concepts such as *word* and traditional IR operators (instead of strings and their operators). Our data models and query languages are more expressive than the one used in the centralized SDI system SIFT [66] where documents are free text and queries are conjunctions of keywords. On the other hand, we have only considered notifications that have *flat* structure, thus our query languages are weaker than the XML-based query languages of [3, 15]. Notice that the similarity concept of $\mathcal{AWPS}$ has also been used in database systems with IR influences (e.g., WHIRL [19]) and more recently in the XML query language ELIXIR [18]. We note that both WHIRL and ELIXIR target information retrieval and integration applications, and pay no attention to information dissemination and the concepts/functionality needed in such applications. Finally, notice that although our data models deal with *textual information* only, they can be easily combined with models for expressing other kinds of information (e.g., arithmetic information for modelling prices in an e-commerce application).

4.2 Algorithms and Computational Complexity

In [37, 41, 38, 39] we point out that in order to have a scalable implementation of a P2P SDI architecture like the one in Figure 1, the following four algorithmic problems need to be formalized and solved efficiently (for a start!). The first problem is the *satisfiability problem*: deciding whether a profile can ever be satisfied by an incoming notification. The second problem is the *satisfaction problem*: Deciding whether a notification satisfies (or matches) a profile. The third problem (which includes the second one) is the *filtering problem*: Given a database of profiles db and a notification n, find all profiles $q \in db$ that match n. This functionality is very crucial at each middle-agent because we expect deployed information dissemination systems to handle hundreds of thousands or millions of profiles. The fourth problem is the *entailment problem*: Deciding whether a profile is more or less "general" than another. This functionality is crucial if we want to minimize profile forwarding as sketched above.

[41] presented PTIME worst-case upper bounds for the complexity of satisfaction and filtering for the case of models $\mathcal{WP}$ and $\mathcal{AWP}$. In [39] we extend these results and study the computational complexity of query satisfiability and entailment for models $\mathcal{WP}$ and $\mathcal{AWP}$. Our results show that the satisfiability and entailment problems for queries in $\mathcal{WP}$ is NP-complete and coNP-complete respectively. Our results for $\mathcal{AWP}$ show that even for queries in some "canonical" form, satisfiability is NP-complete and query entailment is coNP-complete.

4.3 Implementation

In [38] our group has proposed DIAS, a distributed information alert system for digital libraries that follows the architecture of Figure 1 and employs conjunctive queries in $\mathcal{AWPS}$. Work on DIAS has resulted in the implementation of a prototype *super-peer* system called P2P-DIET that combines ad-hoc querying as found in other P2P networks and SDI as discussed above [33]. P2P-DIET has been developed on top of the mobile agent system DIET Core presented in [32]. P2P-DIET is currently available only to partners of project DIET but we plan to make it freely available to all interested parties as soon as the underlying platform DIET Core becomes open source.

4.4 Performance and Scalability

Let us now briefly discuss the data structures, algorithms and protocols that regulate how P2P-DIET nodes work together so that all published notifications are delivered to interested consumers. Our implementation uses a combination of the following techniques: *sophisticated routing* of profiles and notifications by utilizing the reverse path forwarding algorithm of [20] and a poset data structure encoding profile entailment, and *very fast indexing* at each node for detecting the set of profiles that match an incoming notification. Preliminary evaluation of these techniques by us (and previously by SIENA researchers [12]) show that P2P-DIET is a scalable super-peer system that goes beyond the ones envisioned in [67].

In [39] we study the problem of filtering for model $\mathcal{AWP}$ and present a simple but efficient profile indexing algorithm called PIndex. PIndex utilises a two-level index over profiles expressed in $\mathcal{AWP}$ and its detailed data structures are discussed in [60]. We have tested PIndex in a digital library SDI application using documents downloaded from *ResearchIndex*[19] and realistic profiles, consisting of terms extracted from the documents. As it is shown in Figure 2, PIndex can deal with 2.5 million profiles in less than 500 milliseconds and compares very favourably with the obvious brute-force algorithm BF. In our experiments profiles and documents are stored in main memory and time measured is the mean matching time for one hundred incoming documents. The experiments were run on a standard PC with Pentium III 1.6GHz processor and 1GB RAM, running Linux. In recent work we have extended these algorithms to the model $\mathcal{AWPS}$ and the results appear in a forthcoming paper.

5 Conclusions

We have pointed out that the basic concepts underlying P2P systems and MAS are very similar and made a case for a synthesis of the respective research agendas that would benefit both areas. To drive our point home, we presented some

[19] `http://www.researchindex.org`

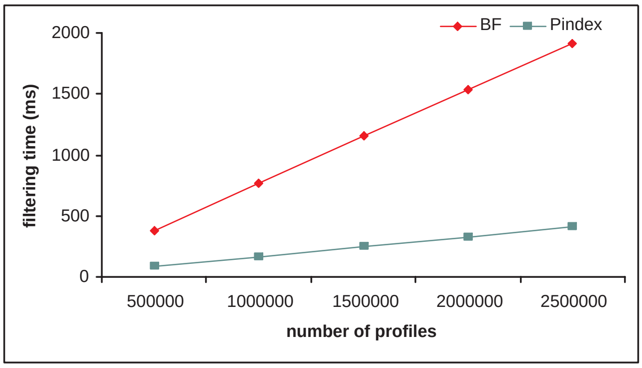

Fig. 2. Matching time vs. number of indexed profiles for $\mathcal{AWP}$ queries

of our work in the area of MAS and P2P that starts from SDI scenarios studied in traditional MAS research and tackles them with middle-agent networks implemented with state-of-the-art techniques from P2P systems.

Acknowledgements

I would like to thank the organizers of CIA-2003 for giving me the opportunity to present my ideas on P2P Computing and MAS.

The research presented here was carried out as part of the DIET (Decentralised Information Ecosystems Technologies) project (IST-1999-10088), within the Universal Information Ecosystems initiative of the Information Society Technology Programme of the European Commission. I would like to thank the other participants of the DIET project and especially my students at TUC for their comments and contributions. Lots of thanks to Thodoros Koutris and Christos Tryfonopoulos whose experimental work resulted in the graph of Figure 2 (and many more!).

References

1. K. Aberer, P. Cudre-Mauroux, and M. Hauswirth. The Chatty Web: Emergent Semantics Through Gossiping. In *Twelfth International World Wide Web Conference (WWW2003)*, May 2003.
2. Eytan Adar and Bernardo A. Huberman. Free Riding on Gnutella. *First Monday*, October 2000.
3. M. Altinel and M.J. Franklin. Efficient filtering of XML documents for selective dissemination of information. In *Proceedings of the 26th VLDB Conference*, 2000.
4. Fred S. Annexstein, Kenneth A. Berman, and Mihajlo A. Jovanovic. Latency Effects on Reachability in Large-scale Peer-to-Peer Networks. In *Proceedings of ACM Symposium on Parallel Algorithms and Architectures*, pages 84–92, 2001.

5. O. Babaoglu. BISON: Biology-inspired techniques for self-organization in dynamic networks, November 2002. Presentation at the launch meeting of the Network of Excellence on Complex Systems (EXYSTENCE) in Torino, Italy.
6. Ozalp Babaoglu, Hein Meling, and Alberto Montresor. Anthill: A framework for the development of agent-based peer-to-peer systems. In *Proceedings of IEEE International Conference on Distributed Computer Systems*, pages 15–22, 2002.
7. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.
8. D. Bertolini, P. Busetta, A. Molani, M. Nori, and A. Perini. Designing peer-to-peer applications: an agent-oriented approach. In *Proceeding of International Workshop on Agent Technology and Software Engineering (AgeS)-Net Object Days 2002 (NODe02)*, volume 2592 of *Lecture Notes in Artificial Intelligence*, pages 1–15. Springer, October 7–10 2002.
9. Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* SFI Studies in the Sciences of Complexity. Oxford University Press, 1999.
10. Erwin Bonsma and Cefn Hoile. A distributed implementation of the SWAN lookup system using mobile agents. In *Proceedings of the AAMAS 2002 Workshop on Agents and Peer-to-Peer Computing*, 2002. Forthcoming volume in Lecture Notes in Computer Science, G. Moro and M. Koubarakis (eds.).
11. A. Campailla, S. Chaki, E. Clarke, S. Jha, and H. Veith. Efficent filtering in publish-subscribe systems using binary decision diagrams. In *Proceedings of the 23rd International Conference on Software Engineering*, Toronto, Ontario, Canada, 2001.
12. A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Achieving scalability and expressiveness in an internet-scale event notification service. In *Proceedings of the 19th ACM Symposium on Principles of Distributed Computing (PODC'2000)*, pages 219–227, 2000.
13. Antonio Carzaniga, David S. Rosenblum, and Alexander L Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383, August 2001.
14. J. Castro, M. Kolp, and J. Mylopoulos. Towards requirements-driven software development methodology: The Tropos project. *Information Systems*, 27(6):365–389, June 2002.
15. C.-Y. Chan, P. Felber, M. Garofalakis, and R. Rastogi. Efficient Filtering of XML Documents with XPath Expressions. In *Proceedings of the 18th International Conference on Data Engineering*, pages 235–244, February 2002.
16. C.-C. K. Chang, H. Garcia-Molina, and A. Paepcke. Boolean Query Mapping across Heterogeneous Information Sources. *IEEE Transactions on Knowledge and Data Engineering*, 8(4):515–521, 1996.
17. C.-C. K. Chang, H. Garcia-Molina, and A. Paepcke. Predicate Rewriting for Translating Boolean Queries in a Heterogeneous Information System. *ACM Transactions on Information Systems*, 17(1):1–39, 1999.
18. T. T. Chinenyanga and N. Kushmerick. Expressive retrieval from XML documents. In *Proceedings of SIGIR'01*, September 2001.
19. William W. Cohen. WHIRL: A word-based information representation language. *Artificial Intelligence*, 118(1-2):163–196, 2000.
20. Y. Dalal and R. Metcalfe. Reverse Path Forwarding of Broadcast Packets. *Communications of the ACM*, 21(12):1040–1048, 1978.

21. N. Daswani, H. Garcia-Molina, and B. Yang. Open problems in data sharing peer-to-peer systems. In *Proceedings of the 9th International Conference on Database Theory (ICDT 2003)*, volume 2572 of *Lecture Notes in Computer Science*, pages 1–15. Springer, January 2003.
22. K. Decker, K. Sycara, and M. Williamson. Middle-agents for the internet. In *Proceedings of IJCAI-97*, 1997.
23. J. Ferber. *Multi-Agent Systems: An introduction to Distributed Artificial Intelligence*. Addison-Wesley, 1999.
24. T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an Agent Communication Language. In N. Adam, B. Bhargava, and Y. Yesha, editors, *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, pages 456–463, Gaithersburg, MD, USA, 1994. ACM Press.
25. T.W. Finin and Y. Labrou. Napster as a Multi-Agent System. Presentation at the 18th FIPA meeting, University of Maryland Baltimore County, July 2000.
26. Gary William Flake. *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation.* MIT Press, 2000.
27. M. J. Franklin and S. B. Zdonik. "Data In Your Face": Push Technology in Perspective. In *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 516–519, 1998.
28. A. Galardo-Antolin, A. Navia-Vasquez, H.Y. Molina-Bulla, A.B. Rodriquez-Gonzalez, F.J. Valvarde-Albacete, A.R. Figueiras-Vidal, T. Koutris, A. Xiruhaki, and M. Koubarakis. I-Gaia: an Information Processing Layer for the DIET Platform. In *Proceedings of the 1st International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2002)*, pages 1272–1279, July 15–19 2002.
29. Bugra Gedik and Ling Liu. PeerCQ: A Decentralized and Self-Configuring Peer-to-Peer Information Monitoring System. In *Proceedings of the 23rd IEEE International Conference on Distributed Computer Systems*, May 2003.
30. Fausto Giunchiglia and Ilya Zaihrayeu. Making peer databases interact - a vision for an architecture supporting data coordination. In *Proceedings of the 6th International Workshop on Cooperative Information Agents (CIA 2002)*, volume 2446 of *Lecture Notes in Computer Science*. Springer, September 2002.
31. Alon Y. Halevy, Zachary G. Ives, Dan Suciu, and Igor Tatarinov. Schema Mediation in Peer Data Management Systems. In *Proceedings of the 19th International Conference on Data Engineering*, March 2003.
32. Cefn Hoile, Fang Wang, Erwin Bonsma, and Paul Marrow. Core specification and experiments in DIET: a decentralised ecosystem-inspired mobile agent system. In *Proceedings of the 1st International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2002)*, pages 623–630, July 15–19 2002.
33. S. Idreos and M. Koubarakis. P2P-DIET: A Query and Notification Service Based on Mobile Agents for Rapid Implementation of P2P Applications. Technical report, Intelligent Systems Laboratory, Dept. of Electronic and Computer Engineering, Technical University of Crete, June 2003.
34. Mihajlo A. Jovanovic. Modeling Large-scale Peer-to-Peer Networks and a Case Study of Gnutella. Master of Science, University of Cincinnati, June 2001.
35. Greg Karvounarakis, Sofia Alexaki, Vassilis Christophides, Dimitris Plexousakis, and Michel Scholl. RQL: A Declarative Query Language for RDF. In *Proceedings of the 11th International World Wide Web Conference.* May 2002.
36. Anastasios Kementsietsidis, Marcelo Arenas, and Renee J. Miller. Mapping Data in Peer-to-Peer Systems: Semantics and Algorithmic Issues. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, June 2003.

37. M. Koubarakis. Boolean Queries with Proximity Operators for Information Dissemination. Proceedings of the Workshop on Foundations of Models and Languages for Information Integration (FMII-2001), Viterbo, Italy , 16-18 September, 2001. In LNCS (forthcoming).
38. M. Koubarakis, T. Koutris, P. Raftopoulou, and C. Tryfonopoulos. Information Alert in Distributed Digital Libraries: The Models, Languages and Architecture of DIAS. In *Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2002)*, volume 2458 of *Lecture Notes in Computer Science*, pages 527–542, September 2002.
39. M. Koubarakis and C. Tryfonopoulos. Peer-to-peer agent systems for textual information dissemination: algorithms and complexity, December 2002. In the *UK Workshop on Multiagent Systems (UKMAS-2002)*, Liverpool, UK.
40. M. Koubarakis, C. Tryfonopoulos, and Y. Labrou. Agent middleware: A survey, 2003. Forthcoming paper.
41. M. Koubarakis, C. Tryfonopoulos, P. Raftopoulou, and T. Koutris. Data models and languages for agent-based textual information dissemination. In *Proceedings of 6th International Workshop on Cooperative Information Systems (CIA 2002)*, volume 2446 of *Lecture Notes in Computer Science*, pages 179–193, September 2002.
42. D. R. Kuokka and L. P. Harada. Issues and extensions for information matchmaking protocols. *International Journal of Cooperative Information Systems*, 5(2-3):251–274, 1996.
43. B. Langley, M. Paolucci, and K. Sycara. Discovery of infrastructure in multi-agent systems. In *Proceedings of the Second Workshop on Infrastructure for Agents, MAS and Scalable MAS*, 2001.
44. P. Marrow, M. Koubarakis, R.H. van Lengen, F. Valverde-Albacete, E. Bonsma, J. Cid-Suerio, A.R. Figueiras-Vidal, A. Gallardo-Antolin, C. Hoile, T. Koutris, H. Molina-Bulla, A. Navia-Vazquez, P. Raftopoulou, N. Skarmeas, C. Tryfonopoulos, F. Wang, and C. Xiruhaki. Agents in Decentralised Information Ecosystems: The DIET Approach. In M. Schroeder and K. Stathis, editors, *Proceedings of the AISB'01 Symposium on Information Agents for Electronic Commerce, AISB'01 Convention*, pages 109–117, University of York, United Kingdom, March 2001.
45. Alberto Montresor, Ozalp Babaoglu, and Hein Meling. Towards adaptive, resilient and self-organizing peer-to-peer systems. In *Web Engineering and Peer-to-Peer Computing, NETWORKING Workshops 2002, Pisa, Italy, May 19-24*, number 2376 in Lecture Notes in Computer Science, pages 300–305. Springer, 2002.
46. Alberto Montresor, Hein Meling, and Ozalp Babaoglu. Messor: load balancing through a swarm of intelligent agents. In *Proceedings of the AAMAS 2002 Workshop on Agents and Peer-to-Peer Computing*, 2002. Forthcoming volume in Lecture Notes in Computer Science, G. Moro and M. Koubarakis (eds.).
47. W. Neidl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch. EDUTELLA: A P2P networking infrastructure based on RDF. In *Proceedings of the 11th International World Wide Web Conference.* May 2002.
48. Wee Siong Ng, Beng Chin Ooi, and Kian Lee Tan. BestPeer: A Self-Configurable Peer-to-Peer System . In *Proceedings of the 18th International Conference on Data Engineering*, February 2002.
49. M. Paolucci, T. Kawamura, T.R. Payne, and K. Sycara. Semantic Matching of Web Services Capabilities. In *Proceedings of the 1st International Semantic Web Conference (ISWC2002)*, volume 2342 of *LNCS*, pages 333–347, 2002.

50. P. R. Pietzuch and J. Bacon. Hermes: A Distributed Event-Based Middleware Architecture. Proceedings of the International Workshop on Distributed Event-Based systems (DEBS'02), July 2-3, 2002, Vienna, Austria.
51. M. Resnick. *Turtles, Termites and Traffic Jams: Explorations into Massively Parallel Microworlds*. MIT Press, 1994.
52. S. Saroiu, K. Gummadi, and S. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Conferencing and Networking*, January 2002.
53. O. Shehory. A Scalable Agent Location Mechanism. In *Proceedings of ATAL 1999*, pages 162–172, 1999.
54. I. Stoica, D. Adkins, S. Ratnasamy, S. Shenker, S. Surana, and S. Zhuang. Internet indirection architecture. In *Proceedings of ACM SIGCOMM'02*, pages 73–86, August 2002.
55. Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the ACM SIGCOMM '01 Conference*, San Diego, California, August 2001.
56. K. Sycara, M. Klusch, S. Widoff, and J. Lu. Dynamic Service Matchmaking Among Agents in Open Information Environments. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 28(1):47–53, 1999.
57. K. Sycara, M. Paolucci, M. van Velsen, and J. Giampapa. The RETSINA MAS Infrastructure. *Autonomous Agents and Multi-Agent Systems*, 7(1/2), July 2003.
58. K. Sycara, S. Widoff, M. Klusch, and J. Lu. LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. *Autonomous Agents and Multi-Agent Systems*, 5:173–203, 2002.
59. P. Triantafillou, C. Xiruhaki, M. Koubarakis, and N. Ntarmos. Towards high-performance peer-to-peer content and resource sharing systems. In *Proceedings of the First Biennial Conference on Innovative Data Systems Research (CIDR 2003)*, January 2003.
60. C. Tryfonopoulos and M. Koubarakis. Agent-based textual information dissemination: Data models, query languages, algorithms and computational complexity. Technical report, Intelligent Systems Laboratory, Dept. of Electronic and Computer Engineering, Technical University of Crete, October 2002.
61. Mitchell M. Waldrop. *Complexity: The Emerging Science at the Edge of Order and Chaos*. Touchstone Books, 1993.
62. F. Wang. Self-organising Communities Formed by Middle Agents. In *Proceedings of the 1st International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2002)*, pages 1333–1339, July 15–19 2002.
63. H. Chi Wong and K. Sycara. A Taxonomy of Middle-Agents for the Internet. In *Proceedings of 4th International Conference on Multi Agent Systems (ICMAS-2000)*, pages 465–466, Boston, Massachusetts, USA, July 2000.
64. M. Wooldridge, N. R. Jennings, and D. Kinny. The Gaia Methodology for Agent-Oriented Analysis and Design. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.
65. Michael Wooldridge and Nicholas R. Jennings. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.
66. T.W. Yan and H. Garcia-Molina. The SIFT information dissemination system. *ACM Transactions on Database Systems*, 24(4):529–565, 1999.
67. B. Yang and H. Garcia-Molina. Designing a super-peer network. In *Proceedings of the 19th International Conference on Data Engineering (ICDE 2003)*, March 5–8 2003.

68. Franco Zambonelli and H. Van Dyke Parunak. From design to intention: signs of a revolution. In *Proceedings of the 1st International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2002)*, pages 455–456, July 15–19 2002.