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., SETI@Home\(^6\) 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
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 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 \textit{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 \textit{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 \textit{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 \textit{selective dissemination of information} (SDI). For more details the interested


\(^{10}\) 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.imagc.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-active-ness [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.\textsuperscript{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

\textsuperscript{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\textsuperscript{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

\textsuperscript{13} http://www.musicity.com
and take advantage of recently developed Semantic Web technologies. 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 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 TTIs 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 as maximum number or reliability 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

http://www.w3.org/2001/sw/

http://www.jxta.org
a local list of other agents and their capabilities). [33] 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, Shethory 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\(^{10}\) [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 behaviour 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].

\(^{10}\) 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.
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.\footnote{SIENA uses the term server instead of middle-agent but the idea is exactly the same.} 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.
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 $WP$, $AVP$ and $AVPS$, and their corresponding languages for specifying queries and notifications. $WP$ is based on free text and its query language is based on the boolean model with proximity operators. The concepts of $WP$ extend the traditional concept of proximity in IR [7,
in a significant way, and utilize it in a query language targeted at SDI for distributed digital libraries. Data model AVP is based on attributes with finite-length strings as values. Its query language is an extension of the query language of data model WP. Finally, the model AVPS extends AVP 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 AVPS 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 ∈ 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 WP and AVP. In [39] we extend these results and study the computational complexity of query satisfiability and entailment for models WP and AVP. Our results show that the satisfiability and entailment problems for queries in WP is NP-complete and coNP-complete respectively. Our results for AVP show that even for queries in some “canonical” form, satisfiability is NP-complete and query entailment is coNP-complete.
5 Conclusions

We have pointed out that the basic concepts underlying P2P systems and MMS 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 RAM experiments run on a standard PC with Pentium III 667 MHz processor and 1 GB of RAM running Linux. In recent work, we have extended these algorithms to the model MWP and the results appear in a forthcoming paper.

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 of the following techniques: efficient protocols, encoding algorithms, and optimizations have been successfully used, including the nearest-neighbor, encoding method. The optimized algorithms for this set of profiles that match an incoming notification. Preliminary evaluation of the two-level indexing algorithm and the efficient protocols is discussed in [7]. We have used P2P-Diet in a digital library with 1,000 million profiles and 12GB of RAM. As it is shown in Figure 2, RxP-Diet can deal with 2.3 million profiles in less than 30 milliseconds per match. This is the mean matching time for one hundred incoming documents. The experiments were run on a standard PC with Pentium III 667 MHz processor and 1 GB of RAM running Linux. In recent work, we have extended these algorithms to the model MWP and the results appear in a forthcoming paper.

4.3 Implementation

In [38], our group has proposed DAS, a distributed information alert system for active libraries that follows the architecture of Figure 1 and enables efficient querying of library databases in the library system. Work on DAS is currently available only to partners of project DIT-Care. As the platform for making it freely available, we have been investigating platform DIT-Care. Open source
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


