# P2P Evolution: From File-sharing to Decentralized Workflows

## P2P Entwicklung: Vom Filesharing zu dezentralisierten Workflows

Thomas Risse, Predrag Knežević, Andreas Wombacher, Fraunhofer IPSI, Darmstadt, Germany

**Summary** Peer-to-peer (P2P) computing is currently receiving a great deal of attention. Much of the current attention is focused on the use of P2P computing for exchanging media files. But P2P systems have actually been around for a long time and many application scenarios exist. In this paper we discuss on one hand the evolution of P2P middleware from simple file-sharing to decentralized data storage; on the other hand the impact on the future development of decentralized workflows. Building up a P2P based data storage, with similar properties as a standard data store, opens many research issues. While P2P databases enable distributed data storage, distributed control flows are considered in decentralized workflows. The paper will present several research issues addressing the collaboration and coordination problem in distributed environments without a centralized coordination instance by means of a B2B application scenario. ►►► **Zusammenfassung** Peer-to-Peer (P2P) Computing ist derzeit in aller Munde. Das Augenmerk liegt vor allem auf dem Einsatz von P2P Systemen für den Austausch von Medien-Daten. Doch P2P Systeme gibt es schon seit langer Zeit mit verschiedenen Anwendungsszenarien. In diesem Artikel wird einerseits die Entwicklung der P2P Middleware vom Filesharing System zu dezentralisierten Datenspeichern betrachtet und auf der anderen Seite der Einfluss von P2P Systemen auf die zukünftige Entwicklung von dezentralisierten Workflows untersucht. Die Entwicklung dezentralisierter Datenspeicher/Datenbanksysteme mit ähnlichen Eigenschaften wie traditionelle Systeme wirft viele Forschungsfragen auf. Während P2P Datenbanken die verteilte Datenhaltung ermöglichen, werden verteilte Kontrollflüsse in P2P Workflows betrachtet. Der vorliegende Beitrag präsentiert offene Forschungsfragen, die die Probleme der Zusammenarbeit und Koordination in verteilten Umgebungen ohne eine zentralisierte Koordinationsinstanz anhand eines B2B Anwendungsbeispiels behandeln.

**KEYWORDS** H.2.8 [Database Applications] Workflow Management, Distributed databases, Decentralization

### 1 Introduction

The idea of peer-to-peer (P2P) computing dates back to the early stages of the Internet. The first applications that used P2P patterns were Usenet and DNS. Even today, Usenet is one of the most popular Internet services and handles an enormous amount of data, thus showing how powerful its decentralized architecture is. Also DNS has shown its scalability by growing from a few thousand hosts at the beginning to hundreds of millions of hosts nowadays.

There are several of the definitions of P2P that are being used by the P2P community. An appropriate comes from the Intel P2P working group: "P2P is the sharing of computer resources and services by direct exchange between systems" [1]. In P2P environments, systems are no longer distinguished by thin clients and thick servers. i.e., every node (peer in P2P terminology) has, a priori, an equal status. This means that a peer offers services or resources to the community, but at the same time, it can consume services/resources from others in the system. An important property of P2P systems is the lack of a central administration.

These properties make P2P file sharing systems most popular. But the P2P development is moving from simple file sharing to large scale decentralized and reliable systems. A lot of new applications, e.g., reliable sensor networks (distributed data), ambient intelligence (distributed knowledge), and Ubiquitous Computing (distributed and highly interacting mobile devices) will benefit from this development.

It is likely that evolving P2P systems will technologically be based on service oriented computing. The benefit of services is that they are self-contained and self-described. This forms the basis for the future establishment of loosely coupled cross-organizational collaborations without relying on predefined global workflow. The instantiation of decentralized workflows requires new methods for the discovery of trading partners as well as a decentralized soundness checking of the constructed ad-hoc collaborations.

In this paper a potential evolution of P2P systems is sketched starting from current P2P systems and adding more powerful data management functionalities resulting in P2P data stores. Extending the functionality of peers to support state dependent services results in P2P workflow management systems.

The structure of the paper is as follows. In the next Section we give a motivation of the open issues by a concrete business scenario. Afterwards in Section 3 the P2P evolution from file sharing to decentralized workflows is presented. Section 4 discusses related activities and Section 5 gives conclusions and outlook on future work.

## 2 Motivation

The exemplary scenario used for further discussion is a simple procurement workflow within a virtual enterprise incorporating a buyer, an accounting department, and a logistics department. The accounting department checks orders (*order* message) of buyers and forwards them to the logistics department (*deliver* message) to deliver the requested goods. The logistics department confirms the receipt (*deliver_conf* message), which is forwarded by the accounting department to the buyer (*delivery* message). Further, the buyer may perform parcel tracking (*get_status* and *status* messages) as sometimes offered by logistics companies. The overall scenario is depicted in Fig. 1.

Fig. 1 represents the global relationship, but not the local workflows of the parties involved. While different models have been proposed to model workflows, in the following, the *Workflow Net model* (WF-Net) [2] is used due to the support of operations required further on. Other notations as for example, Petri Nets [3], flowcharts [4], or statecharts [5; 6] could also be used. A Workflow Net consists of places (circles) representing business tasks and transitions (squares) connecting places representing a message exchange. The transitions are labelled with *s#r#msg* representing sender *s* and receiver *r* of the message as well as its message name *msg*. WF-Nets contain a single final place represented by a circle with a solid line within the graph. A token is depicted as a dot within a circle. A transition is enabled if all input places of a transition contain a token. If a transition is enabled, it might fire, that is removing tokens from incoming places and inserting new tokens to outgoing places of the transition. The current distribution of tokens over the places describes the actual status of the workflow and is named marking.

The decentralized/local workflows of the different parties are depicted in Fig. 2, which are derived from the global workflow, e. g., by using [7], and guaranteeing a successful execution of the process. The process is started by the buyer *b* sending a *b#a#order* message to the accounting department *a*, which informs the logistics department *l* via a *a#l#deliver* message to deliver the ordered goods. The logistics department *l* confirms this request (*l#a#deliver_conf* message) to the accounting department *a*, which forwards the delivery details of the order (*a#b#delivery* message) to the buyer *b*. Afterwards, the buyer *b* is allowed to do parcel tracking directly with the logistics department *l* by sending a *b#l#get_status* message answered by a *l#b#status* message. Finally, the buyer and logistics department process is terminated by the buyer *b* sending a *b#l#terminate* message.

Within this example the local workflows are created by a top-down approach. Hence the local workflows are derived from the commonly agreed global workflow. The main advantage of the top-down approach is the possibility to easily ensure that the different parties are able to interact with each other and that the global workflow guarantees successful business interactions further called soundness of the multi-lateral collaboration. Within loosely coupled systems the multi-lateral collaborations are established by composing several pre-existing local workflows due to the lack of possibility to manually negotiate the global view of the collaboration. This is because of the ad-hoc nature of the collaboration, thus the non-predictability of the potential partners, e. g., in a Ubiquitous Computing environment with highly interacting mobile devices. Further, the lack of trust be-

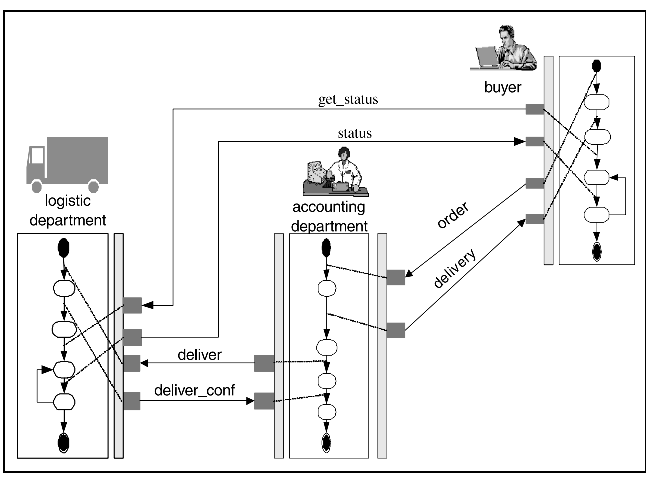


**Figure 1** Global Procurement Scenario.

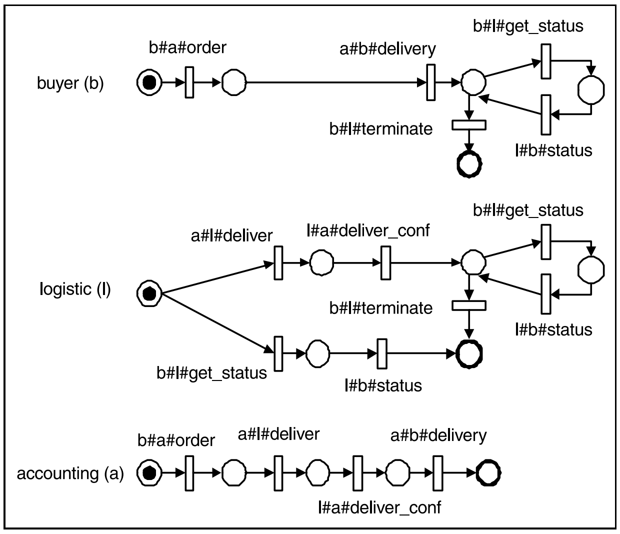


**Figure 2** Local Petri Net Models.

tween the ad-hoc involved parties prevents a soundness check of the collaboration on a global level, but requires each party checking locally the consistency of his local workflow with the related parties. In particular, the decentralized establishment of a multi-lateral collaboration requires

- **Discovery of potential trading partners**
  Discovery of potential trading partners that are providers of a local workflow is based on syntactical, semantical QoS processes, and/or SLA descriptions published by the provider of the local workflow in current service oriented architectures. The description information is maintained by a repository supporting appropriate matchmaking definitions. Such a repository usually is realized either by a decentralized, locally managed but interconnected repositories (e.g., WSIL [8]) or by several centralized but replicated repositories (e.g. UDDI [9]).
- **Decentralized soundness checking of the constructed multi-lateral collaboration**
  Decentralized soundness checking relies on the local workflow models, the bilateral interactions, and a protocol to ensure that each party involved is informed about the successful creation or failure of the multi-lateral collaboration in accordance to the determined soundness of the un-instantiated global workflow.

## 3 P2P Evolution

Service discovery queries are usually more complex than simple name matching, so the service directories would become a bottleneck, when the number of service discovery requests increases. Also, if some network links are down or overloaded, the service directory could be unaccessible to a large portion of potential partners, making them unable to operate. One should not forget administrative costs needed for maintaining such directories.

Within the course of P2P evolution as we expect it to happen in the next 5 years, an infrastructure supporting decentralized discovery and soundness checking has to be provided. The next subsections illustrate the current state and the discusses aspects of the evolution steps.

### 3.1 P2P Current State

Current P2P systems can be roughly divided into following categories: resource-sharing and collaboration. P2P systems got into focus with the file-sharing system Napster, and then followed by Gnutella, KaZaA, and eDonkey [1], where every peer offers its files for community download. They are content-sharing application, a resource-sharing subgroup.

Content-storing applications have taken a slightly different approach for enabling file access for the given community. Every peer offers its disk space that is used by other peers for storing their files. Usually, these systems have embedded privacy and security control. FreeNet, GNUNet, Past, or OceanStore [1] are just a few example for content-storing systems.

P2P collaboration being quite close to everyday human communication, which is usually direct. Good examples are messenger applications like ICQ, Jabber, YahooMessenger, Groove [1], and recently Skype [10], an P2P IP telephony solution.

The main issue in the presented content-sharing and content-storing systems is to make content available, but how to lookup the appropriate files/objects. Napster, which is a hybrid P2P system, uses a centralized index that is used for location of files. Gnutella as the first pure P2P system uses flooding for sending the query into the community. Such approach is not very efficient and scalable [11; 12].

Having this in mind, further research in P2P domain proposed *Distributed Hash Tables* (DHT) as the next generation of low-level P2P systems. DHTs are quite mature and many implementations like Pastry, Tapestry, P-Grid, CAN, and Chord [1] are available. DHT provides functionalities of a hash table data structure distributed in the network. The core of each DHT is its routing algorithm that delivers messages to appropriate nodes. It is scalable and resistant to dynamic changes in the network (e.g., community size and memberships, network topologies). DHT systems are used as ground layers of P2P applications (e.g. PAST, OverNet [13]).

The issues like reputation, trust, schema interoperability, efficient querying are still subjects of ongoing research and therefore, some of the presented systems do not have pure P2P design, e.g., they use a central place for user authorization and authentication, or they rely on the presence of some peers.

Current P2P file-sharing systems seem to be a good starting point for enabling decentralized service discovery. Every service would publish its service description as a file available for the whole community. In theory, that would make a decentralized service description repository. However, these systems have some important drawbacks: file-level granularity and write-once access. File-level granularity means that the query capabilities are restricted to file names. So, it is not possible to search for a particular content inside files, which is required for more complex matchmaking operations. Any advanced searching mechanism like qualified, range or boolean predicates search are not supported by current systems. Furthermore, as the files are non-updateable after storing, maintenance of the service descriptions is not supported in an appropriate way.

There are some attempts [14] to extend Gnutella protocols to support other types of queries. It would be quite possible to make a Gnutella implementation that understands some variant of SQL, XPath, or XQuery. However, such networks would have problems with system load, scalability and data consistency, e.g., only locally stored data could be updated and mechanisms updating other replicas do not exist.

In order to have better update and query capabilities, a decentralized data store is required supporting fine-grained object handling. The store requirements are similar to those explored in the centralized and distributed databases (Sect. 4.1).

Details about additional motivating scenarios and their support by decentralized datastores are presented in [15].

### 3.2 Decentralized Data Management

As a consequence of the limitations of current P2P systems a decentralized service discovery requires a P2P data store that manages semi-structured data. Semi-structured data, like XML, is very often used for data exchange in heterogeneous systems, but also for meta-information like service descriptions. The requested P2P data store differs from the systems discussed in the previous section in a way that the data is split into finer pieces, which are spread afterward within the community. They are created and modified by the community during system run-time and they can be accessed from any peer in a uniform way, e.g., a peer does not have to know anything about the data allocation.

In order to realize a decentralized service description repository with advanced query mechanisms, the community of service providers will create and maintain in the P2P data store a pool of service descriptions. Every service will be able to modify its description during the lifetime and to search for needed services. Query execution will be spread at many peers, so the query originator will only obtain the final result.

Building a reliable decentralized data store would require at least attention to the following issues: database design, directory management, concurrency control, and reliability. The field of distributed databases (DDBS) [16] gives a good overview of existing limitations and solutions that could be applied to decentralized data stores as well. Although, distributed databases do not capture node leaving and unknown global structures being a consequence of the highly dynamic of P2P systems. In particular, this vitality makes it very challenging to achieve high data availability, i.e., data must be available even if the creator goes offline. Therefore, the data must be replicated. It is the job of the replica control protocol to find a good trade-off between several system parameters (e.g., peer availability, community size, and required data availability). Known DDBS replica control protocols and algorithms [16; 17] are good starting points for further research. They offer different levels of data consistency (from strict to lazy and none) and support different node availability requirements.

Further analysis will provide the answer, which consistency models can be applied to our datastore. For example, in decentralized service discovery scenario is not really needed that decentralized data store keeps data about services that are off-line. In some other scenarios (e.g., storing requests and corresponding service response in order to re-use them later), decentralized data store should preserve data, even if the author disappears.

A more detailed discussion of this and further challenges and issues can be found in [18].

### 3.3 Decentralized Workflows

The bottom-up creation of a multilateral collaboration is based on service discovery discussed in the previous section and decentralized soundness checking. The latter issue has been rarely addressed by the workflow community so far. The aim is to start with local workflows combining them and decide soundness, that is deadlock-freeness and boundedness, of the resulting global workflow without instantiating the global workflow model in a decentralized way.

An intuitive approach extends the local workflow by the relevant parts of the trading parties' local workflows, which can be derived by neglecting those parts of the trading parties' workflows not being part of this particular bilateral interaction. In particular, those transitions are omitted by relabelling with a silent message, which are neither sent nor received by the corresponding trading partner. In [19] such a definition on WF-Nets is provided known as projection inheritance. Applying this projection definition to the trading partners interaction results in the party's view on the trading partners local workflows. Combining these views and the party's local workflow result in a workflow model used by the party to decide soundness of the global workflow in a decentralized way.

Based on this definition, the soundness decision can be made straightforward in tree like structures, but causes problems in graph structures, where cyclic dependencies exist as contained in the example described in Sect. 2. The reason for the failure in cyclic depended workflows is the information loss introduced by the projection.

An exemplary case illustrating the limitation of the above sketched approach can be observed by detecting a deadlock at the constructed logistics departments workflow, although the exemplary global workflow is sound. In the logistics constructed workflow the *b#l#get_status* and the *a#l#deliver* messages might occur in arbitrary order. This is due to a loss of message order constraint contained on the buyer local workflow, but not visible to the constructed logistics workflow resulting in a deadlock, if the sequence *b#l#get_status l#b#status* is executed. The detected deadlock in the constructed workflow indicates the virtual global workflow not to be sound, although it is sound.

The lesson learned from this example is that the information loss about inherent message ordering constraints in the extended logistics department workflow results in options, which might be negligible due to decisions made by another party.

Another type of information loss generated by the view calculation is related to parameter value constraints. An example can be constructed based on the one in Sect. 2, where the buyer and the logistics department are allowed to perform orders up to a maximum volume, while the accounting department has no limitation on order volumes. Due to the fact that the order is sent by the buyer to the accounting, which forwards it to the logistics, the buyer constraint on the order volume limit is not reflected in the constructed logistics department workflow. Thus, a message sequences starting with a *a#l#deliver* message representing an order with a volume above the logistics and buyers limitation, causes a deadlock at the constructed logistics workflow, although the global workflow is sound. Again, the information loss introduced by the view generation causes this wrong local decision.

The approach sketched above guarantees that only sound global workflows will be accepted, although the approach is to restrictive by discarding some sound global workflows. The reason for introducing wrong results is neglecting parameter values as well as message ordering constraints.

The conclusion of the presented examples is that a valid approach for decentralized decision making on sound multi-lateral collaborations should extend the sketched approach by making constraints explicit and using them locally. In particular, the transitivity property of these constraints can be used. This idea may require additional communication between the trading parties to determine appropriate transitivity chains in case of cyclic dependencies, but applies to both message order as well as parameter value constraints. Such an approach will guarantee decentralized decisions on sound non-instantiated global workflows. A more detailed discussion of the constraints and the resulting approach can be found in [20].

## 4 Related Work

### 4.1 Decentralized Service Discovery

Peer-to-peer systems can be observed as a relaxed variant of the distributed systems. The area of the distributed systems is very well explored, and many available books cover different aspects, e.g., [16; 17; 21]. Usually, distributed systems are built with an important assumption: working in stable, well connected environments (e.g., LANs) with the global system overview, where every crashed node is eventually replaced by a new proper one. Also, they require some sort of administration and maintenance.

Jini [22] is an open architecture that enables developers to create network-centric services. The architecture supports service discovery and it is done through centralized Lookup service.

Many people think that Web services are nothing more than just a reinvention of the approach done by CORBA [23], where service discovery is done by using CORBA Trading Service [24]. Comparing to UDDI, CORBA approach is distributed, but not decentralized, e.g., parties that want to search/offer a service must know the trader's location.

Many distributed databases and filesystems are available on the market. The most representative examples of distributed databases are Teradata, Tandems NonStopSQL, Informix Online Xps, Oracle Parallel Server, and IBM DB2 Parallel Edition [25]. The first successful distributed filesystem was Network File System (NFS) that became a kind of standard very quickly. It was succeed by Andrew File System (AFS) and the other proposals like Coda and xFS [21].

A P2P store as proposed in Sect. 3.2 cannot be built by a direct usage of a running system, but the experience gained in the area of distributed systems is, however, extremely helpful in ongoing research in the peer-to-peer domain, even if some important properties (e.g., system overview, reliable network connections) do not exist.

In addition to the data management issues addressed so far, some service descriptions are referenced here to illustrate the type of required matchmaking operations.

An exemplary logic based approach is Web Service Request Language (WSRL) [26], where the corresponding matchmaking is based on temporal and linear constraint satisfaction. DAML-S [27] aims to describe services in semantic web communities. The main draw back of semantic annotation is the necessity of a common ontology used for annotating and querying services. Unfortunately, no such ontology currently is in place. Process annotations are used in [28] being based on finite state automata, while the corresponding matchmaking operation is defined as non-empty intersection.

The different description approaches might be used to extend the functionality of a classical UDDI repository but could not integrated within it because of the complexity of the matchmaking operations.

### 4.2 Decentralized Establishment of Multi-lateral Collaborations

Decentralized collaboration establishment might be investigated under consideration of several approaches starting with an analysis of classical workflow management theory, speech act theory, and finally coordination theory. Classical workflow systems uses for example workflow interoperability standard [29] or jointFlow [30] requiring a centralized consistency checking of the distributed workflow engines, thus does not support a decentralized decision and execution of multi-lateral collaborations. Contrary to the centralized control flow, coordination theory supports centrally shared data. For example, the WorkSpace [31] approach is based on the notion of steps representing a transformation of one or several data elements. The limitation of these approaches to decentralized decision making for multi-lateral collaborations is the need for centralized data maintenance.

Alternatively, Speech Act Theory (SAT) [32] characterizes the communication between parties by means of the intend of the speaker, the effect on the listener, and the physical manifestations of an utterance [33]. Models being based on SAT are logic based like Knowledge Query and Manipulation Language (KQML) [34], dynamic deontic logic [35], or Courteous Logic Programs [36]. All models have in common that they are very generic, but do not explicitly provide the reasoning capabilities required for doing decentralized multi-lateral collaboration establishment.

## 5 Conclusion

This paper describes our view on the evolution of P2P systems and the possible impact on current service oriented architectures being the basis for the establishment of multi-lateral collaborations based on complex services. Currently, service oriented systems rely on central repositories of service descriptions. Our analysis shows that a full decentralization of service oriented architecture provides its full potential when handling complex discovery processes and soundness checking of collaborations. P2P systems form a good basis for decentralization. Hence, current P2P data management systems have to be extended with decentralized data stores, which provide fine granular query capabilities and guarantee a high availability of data. Furthermore, a decentralized decision making will guarantee successful business interactions within multi-lateral collaborations.

Overall, decentralized multi-lateral collaborations will play an important role in future Enterprise Application Integration (EAI) solutions.

## References


[1] D. Milojičić et al.: Peer-to-peer computing. Technical Report, 2002. http://www.hpl.hp.com/techreports/2002/HPL-2002-57.pdf.

[2] W. van der Aalst, K. van Hee. *Workflow Management – Models, Methods, and Systems*. MIT Press, 2002.

[3] K. Jensen: *Coloured Petri Nets*, Springer Verlag, Heidelberg, 1992.

[4] P. Grefen, K. Aberer, Y. Hoffner, H. Ludwig: Crossflow: Cross-organizational workflow management in dynamic virtual enterprises. *Int'l J. of Computer Systems Science & Engineering*, 15(5):277–290, Sep. 2000.

[5] D. Harel: Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274, June 1987.

[6] A. Peron: Statecharts, transition structures and transformations. In *Proc. Int'l Conf. Colloquium on Trees in Algebra and Programming (CAAP-TAPSOFT'95)*, pp.454–468, 1995.

[7] W. van der Aalst, M. Weske: The P2P approach to interorganizational workflows. In *Proc. of 13th Int'l Conf. on Advanced Information Systems Engeneering (CAISE'01)*, 2001.

[8] K. Ballinger, P. Brittenham, A. Malhotra, W.A. Nagy, S. Pharies: Specification: Web services inspection language (ws-inspection) 1.0, Nov. 2001. http://www.ibm.com/developerworks/library/ws-wsilspec.html.

[9] IBM, Microsoft, HP, Oracle, Intel, and SAP. Universal description, discovery, and integration, July 2002. http://www.uddi.org/.

[10] Skype, 2003. http://www.skype.com.

[11] B. Yang, H. Garcia-Molina: Comparing hybrid peer-to-peer systems. In *Proc. of the 27th Int'l Conf. on Very Large Data Bases*, pp.561–570, 2001.

[12] Why gnutella cannot scale, 2004. http://www.darkridge.com/jpr5/doc/gnutella.html.

[13] Overnet, 2003. http://www.overnet.com.

[14] GPU: A gnutella processing unit, 2004. http://gpu.sf.net.

[15] T. Risse, P. Knežević: Data storage requirements for the service oriented computing. In *SAINT 2003 – Workshop on Service Oriented Computing*, pp. 67–72, Jan. 2003.

[16] M. Tamer Özsu, P. Valduriez: *Principles of Distributed Database Systems*. Prentice Hall, 1999.

[17] P. Bernstein, V. Hadzilacos, N. Goodman: *Concurency Control and Recovery in Database Systems*. Addison-Wesley, 1997.

[18] P. Knežević: Towards a reliable peer-to-peer xml database. In *Proc. ICDE/EDBT PhD Workshop 2004*. Crete University Press, Mar. 2004.

[19] W. van der Aalst: Inheritance of Interorganizational Workflows to Enable Business-to-Business E-commerce. *Electronic Commerce Research*, 2(3):195–231, 2002.

[20] A. Wombacher, K. Aberer: Requirements for workflow modeling in P2P-workflows derived from collaboration establishment. *Submitted to ICWS 04*, 2004.

[21] G. Coulouris, J. Dollimore: *Distributed Systems – Concepts and Design*. Addison Wesley, 1989.

[22] Jini network technology, 2004. http://wwws.sun.com/software/jini/.

[23] A. Gokhale, B. Kumar, A. Sahuguet: Reinventing the wheel? CORBA vs. Web services. In *11th Int'l Conf. on World Wide Web (WWW2002)*, May 2002.

[24] CORBA trading object service specification, 2000. http://www.omg.org/docs/formal/00-06-27.pdf.

[25] L. Brunie, H. Kosch: A communications-oriented methodology for load balancing in parallel relational query processing, 1995.

[26] M. Aiello, M. Papazoglou, J. Yang, M. Pistore, M. Carman, L. Serafini, P. Traverso: A request language for web services based on planning and constraint satisfaction. In *Proc. 3rd Int'l Workshop, Technologies for E-Services (TES)*, pp.76–85, 2002.

[27] A. Ankolekar, M. Burstein, J.R. Hobbs, O. Lassila, D. McDermott, D. Martin, S.A. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara: DAML-S: Web service description for the semantic web, 2002. http://citeseer.nj.nec.com/ankolekar02damls.html.

[28] A. Wombacher, P. Fankhauser, B. Mahleko, E. Neuhold: Matchmaking for business processes based on choreographies. In *Proc. Int'l Conf. on e-Technology, e-Commerce and e-Service (EEE-04)*, 2004. to appear.

[29] Workflow Management Coalition. Workflow standard – interoperability abstract specification. http://www.wfmc.org/standards/docs/TC-1012_Nov_99.pdf, Nov 1999.

[30] OMG. Omg business object domain task force bodtf-rfp 2 submission workflow management facility, jflow – joint rfp submission. http://www.omg.org/docs/bom/97-08-05.pdf, Aug 1997.

[31] R. Tolksdorf: Coordination Technology for Workflows on the Web: Workspaces. In *Proc. 4th Int'l Conf. on Coordination Models and Languages*, pp. 36–50, 2000.

[32] K. Bach, R. Harnish: *Linguistic Communication and Speech Acts*. The MIT Press 1979.

[33] D. Marinescu: *Internet-Based Workflow Management – Towards a Semantic Web*. John Wiley, 2002.

[34] T. Finin, Y. Labrou, J. Mayfield: KQML as an agent communication language. In Jeffrey M. Bradshaw, editor, *Software Agents*, chapter 14, pp. 291–316. AAAI Press / The MIT Press, 1997.

[35] J. Meyer, R. Wieringa, F. Dignum: The role of deontic logic in the specification of information systems. In G. Saake J. Chomicki, editor, *Logics for Databases and Information Systems*. Kluwer, 1998.

[36] B. Grosof, D. Levine, H. Chan, C. Parris, J. Auerbach: Reusable architecture for embedding rule-based intelligence in information agents. In T. Finin and J. Mayfield, editors, *Proc.CIKM'95 Workshop on Intelligent Information Agents*, Baltimore, MD, USA, 1995.


1


2


3

**1 Dipl.-Inf. Thomas Risse** is division manager of the data management group at Fraunhofer IPSI. He obtained in 1997 his Diploma in Computer Science at the Technical University of Darmstadt, Germany. In 1998 he joined the research division OASYS at Fraunhofer IPSI in Darmstadt. Since then he worked in several European and industrial projects. His research interests are distributed data management, especially for peer-to-peer environment, performance modelling, and scheduling.
Address: Fraunhofer IPSI, Dolivostraße 15, 64293 Darmstadt, Germany,
Tel.: +49-341/4912-2701,
E-Mail: risse@ipsi.fraunhofer.de

**2 Dipl.-Ing. Predrag Knežević** graduated at School of Electrical Engineering, University of Belgrade, Serbia and Montenegro. Since December 2001, he is a researcher with OASYS department, FhG IPSI. His main focus are peer-to-peer and decentralized datastores. In particular, he explores solving reliability and consistency within those environments.
Address: Fraunhofer IPSI, Dolivostraße 15, 64293 Darmstadt, Germany,
Tel.: +49-341/4912-2701,
E-Mail: predrag.knezevic@ipsi.fraunhofer.de

**3 Dipl.-Inf. Andreas Wombacher** graduated at the Technical University Darmstadt, Germany. He has worked for two years at IBM on electronic commerce systems before he joined Fraunhofer IPSI in 1998. Since then he worked in several European and industrial projects. His research interests are matchmaking of workflows and decentralized workflow management applied to information commerce as well as eServices.
Address: Fraunhofer IPSI, Dolivostraße 15, 64293 Darmstadt, Germany,
Tel.: +49-341/4912-2701,
E-Mail: wombach@ipsi.fraunhofer.de