A Survey of Semantics-based Approaches for Context Reasoning in Ambient Intelligence

Antonis Bikakis, Theodore Patkos, Grigoris Antoniou, and Dimitris Plexousakis

Institute of Computer Science, FO.R.T.H., Vassilika Voutwn
P.O. Box 1385, GR 71110, Heraklion, Greece
{bikakis,patkos,antoniou,dp}@ics.forth.gr

Abstract. A key issue in the study of Ambient Intelligence is reasoning about context. The aim of context reasoning is to deduce new knowledge, based on the available context data. The endmost goal is to make the ambient services more "*intelligent*"; closer to the specific needs of their users. The main challenges of this effort derive from the imperfect context information, and the dynamic and heterogeneous nature of the ambient environments. In this paper, we focus on semantics-based approaches for reasoning about context. We describe how each approach addresses the requirements of ambient environments, identify their limitations, and propose possible future research directions.

1 Introduction

Pervasive applications aim at providing the right information to the right users, at the right time, in the right place, and on the right device. In order to achieve this, a system must have a thorough knowledge and, as one may say, "*understanding*" of its environment, the people and devices that exist in it, their interests and capabilities, and the tasks and activities that are being undertaken. All this information falls under the notions of *context*.

The need for *reasoning* in context aware systems derives from the basic characteristics of context data. Two of these are *imperfection* and *uncertainty*. Henricksen and Indulska [1] characterize four types of imperfect context information: *unknown*, *ambiguous*, *imprecise*, and *erroneous*. Sensor or connectivity failures result in situations, that not all context data is available at any time. When the data about a context property comes from multiple sources, the context information may become ambiguous. Imprecision is common in sensor-derived information, while erroneous context information arises as a result of human or hardware errors. The role of reasoning in these cases is to detect possible errors, make predictions about missing values, and decide about the quality and the validity of the sensed data. The raw context data needs, then, to be transformed into meaningful information so that it can later be used in the application layer. In this direction, some suitable sets of rules can exploit the real meaning of some raw values of context properties. Finally, context reasoning may play the role of a decision making mechanism. Based on the collected context information, and on a set of decision rules provided by the user, the system can be configured to change its behavior, whenever certain changes are detected in its context.

If we also consider the high rates in which context changes and the potentially vast amount of available context information, the reasoning tasks become even more challenging. Overall, Knowledge Management in Ambient Intelligence should enable: *(a)* Reasoning with the highly dynamic and ambiguous context data; *(b)* Managing the potentially huge piece of context data, in a real-time fashion, considering the restricted computational capabilities of some mobile devices; and *(c)* Collective intelligence, by supporting information sharing, and distributed reasoning between the entities of the ambient environment.

In this paper, we present the various solutions that have been proposed to date, giving more attention to those that employ Semantic Web-based representations to describe context. The use of ontology languages is becoming common in such applications mainly because they offer enough representational capabilities to develop a formal context model that can be shared, reused, extended for the needs of specific domains, but also combined with data originating from other sources. Moreover, the development of the Semantic Web logic layer is resulting in rule languages that will enable reasoning with the user's needs and preferences and with the available ontology knowledge. According to the discussion on *Interactive Context-Aware Systems Interacting with Ambient Intelligence* in [2], ontology-based models manage to satisfy all demands placed concerning context modeling, such as distributed composition, partial validation, richness and quality of information, incompleteness and ambiguity, level of formality and, also, applicability to existing environments.

The rest of the paper is structured as follows: Section 2 focuses on ontological reasoning solutions, and Section 3 on rule-based approaches. Section 4 describes methods and techniques for distributed reasoning, while Section 5 discusses additional reasoning techniques concerning *learning*, *offline reasoning* and *probabilistic reasoning*. The last section proposes future research directions that may lead to more efficient reasoning solutions.

2 Ontological Reasoning

The SW Languages of RDF(S) and OWL are common formalisms for context representation. Along with their evolution, a number of SW Query languages (e.g. RDQL [3], RQL [4], TRIPLE [5]) and reasoning tools (e.g. FaCT [6], RACER [7], Pellet [8]) have been developed. Their aim is to retrieve relevant information, check the consistency of the available data, and derive implicit ontological knowledge. The studies of [9] and [10] describe the use of RDQL for accessing RDF context data, while the Context-Aware Guide described in [11] demonstrates the use of RQL in location-based mobile services. An interesting study that describes and evaluates the use of description logic for both representation and reasoning over context is presented in [12]. Below, we present representative examples of systems that reason with context data using Description Logics.

The P2P-based mobile environment in [13] consists of stations that provide semantic services and users with mobile devices, which manage their owner's semantic profile. Both the semantic services and the users' profiles are modeled as description logic predicates. The semantic matching between the services and the profiles, which determines whether a given profile is semantically compatible to a particular service and, if so, how well both do match, is accomplished by applying a set of DL rules, which are processed by a RACER reasoning engine.

In [14], they use a case study from the smart home domain (specifically a context-aware door-lock) to present their approach for modeling and reasoning about context using Description Logics. They have built an OWL schema to model the required context entities, and test three DL reasoners (RACER, its commercial successor RacerPro [15], and Pellet) using a real-case application scenario. However, their scenario is rather too simple to evaluate the performance of these reasoners in much broader context-aware applications.

The ontological reasoning approaches have two significant advantages. They integrate well with the ontology model, which is widely used for the representation of context; and most of them have relatively low computational complexity, allowing them to deal well with situations of rapidly changing context. However, their limited reasoning capabilities are a trade-off that we cannot neglect. They cannot deal with missing or ambiguous information, which is a common case in ambient environments, and are not able to provide support for decision making. Thus, we argue, that although we can use them in cases where we just want to retrieve information from the context knowledge base, check if the available context data is consistent or derive implicit ontological knowledge, they cannot serve as a standalone solution for the needs of ambient context-aware applications.

3 Rule-based Reasoning

In the Ambient Intelligence domain, rules are primarily used to express policies, constraints and preferences. Below, we present some representative examples.

In the SOCAM architecture, they use FOL rules to reason about context ([16]). To resolve possible conflicts, they have defined sets of rules on the classification and quality information of the context data. They suggest that different types of context have different levels of confidence and reliability. For example, defined context is more reliable compared to sensed and deduced context. They also have different levels of quality; for example, an RFID-based location sensor may have a 80% accuracy rate whereas a Bluetooth-based sensor may only have a 60% accuracy rate. The reasoning engine is implemented in Jena2.

In the *Semantic Space* Architecture, there are two modules for retrieving and deriving new information from the OWL Knowledge Base ([17]). The Context Query Engine provides an interface for applications to extract desired context information from the knowledge base. The Context Reasoner enables the users to deduce higher level knowledge, based on the context data of the KB, using FOL rules. The system uses Jena2 to perform forward-chaining reasoning over the KB, based on the rules provided by the user. The same approach is also followed in the prototype context-aware implementation descibed in [18].

As part of *Gaia*, Ranganathan and Campbell propose a FOL-based context infrastructure ([19]). The context information is represented as first-order predicates, with the name of a predicate being the type of context described. The model allows both universal and existential quantification over variables. This allows parameterizing context and representing a much richer set of contexts. A predefined set of rules is used to deduce higher-level knowledge based on the raw context data. Whenever a change occurs in the system's context, the rules are re-evaluated and the new inferred context replaces the old one. To resolve conflicts that occur when multiple rules are activated in the same time, they have developed a priority base mechanism, allowing only one rule to fire at each time. For the evaluation of the rules, they use the XSB reasoning engine.

In [20], the use of OWL is proposed both for context representation data, and for the rules expressing the user preferences and security constraints. Once all the context knowledge has been loaded in system (implemented on Jess), some predefined forward-chaining rules are used to complete the core knowledge base. The service invocation rules, and the privacy enforcing rules, both represented as backward-chaining rules are then applied to the knowledge base.

The Semantic Context-Aware Access Control Framework in [21] uses a combination of Description Logics and Logic Programming reasoning. Specifically, they define two types of rules: (a) *context aggregation rules* to support reasoning using property path relationships; (b) *context instantiation rules* to provide OWL assertions for attribute values. Both types of rules are expressed according to the following pattern: *if context attributes* $C_1 ... C_n$ *then context attribute* C_m, which corresponds to a Horn clause, where predicates in the head and in the body are represented by classes and properties defined in the context and application-specific ontologies. A similar hybrid reasoning approach is also implemented in the context-aware service adaptation middleware described in [22]).

Rule languages provide a formal model for context reasoning. Furthermore, they are easy to understand and widespread used, and there are many systems that integrate them with the ontology model. However, all these approaches share a common deficiency; they cannot handle the highly changeable, ambiguous and imperfect context information. In many of the cases that we described, they had to build additional reasoning mechanisms to deal with conflicts, uncertainty and ambiguities. The proposed logic models suit better in cases, where we are certain about the quality of the collected data. Consequently, neither of these models can serve as the solution to the required reasoning tasks.

4 Distributed Reasoning Techniques

In an Ambient Intelligence environment, there coexist many different entities that collect, process, and change the context information. Although they all share the same context, they face it from different viewpoints based on their perceptive capabilities, their experiences and their goals. Moreover, they may have different reasoning, storage and computing capabilities; they may "speak" different languages; they may even have different levels of sociality. This diversity raises additional research challenges in the study of smart spaces, which only few recent studies have addressed. In the following paragraphs, we present these approaches, which have the common feature of employing methods and techniques from the field of Distributed Artificial Intelligence.

One such approach is *sTuples* ([23]). This framework extends Tuple Spaces using SW technologies to represent and retrieve tuples from a Tuple Space. The Tuple Space model uses a logically shared memory, where producers add tuples to a common space, while consumers read or extract tuples from the space using a search template. The sTuples model advances the space lookup operations using DAML+OIL for the representation of context entities and RACER as the reasoning engine. It provides a generic framework to implement clients and services in a pervasive environment by using service and data tuples. Data tuples are semantic descriptions of the context data that an entity is willing to share with other entities in the environment, while service tuples are advertisements of the services offered in the same environment. Each entity uses various types of agents to gain access to the Tuple Space, each of which has a distinct role. Examples of such roles are, managing the addition, removal and state changes of tuples, searching in the Tuple Space, recommending services to the user, and notifying the user about tuple changes.

Similar approaches, which combine SW technologies and shared memory models to support asynchronous communications in ambient environments, are the *Semantic Spaces* ([24]), and the context management framework presented in [25]. The latter follows a *blackboard*-based approach. A mobile terminal system uses a central context manager, which stores context information from any available source. Clients can directly query the manager to gain context information, subscribe to various context change notification services, or use higher level contexts transparently. In the latter case, the context manager assigns the reasoning tasks to dedicated recognition services.

The OWL-SF framework ([26]) combines the OMG's *Super Distributed Objects* (SDO) technology and the OWL language to allow the distribution of semantically annotated services for the needs of ambient context-aware systems. SDOs are logical representations of hardware and software entities that are used to enable distributed interoperability. The proposed framework integrates two basic building blocks, OWL-SDOs and Deduction Servers. The OWL-SDOs are semantic extensions of SDOs; they use the OWL language to describe their status, services and communication interface. Deduction servers are specific OWL-SDOs that provide reasoning services. They contain a deduction engine coordinating reasoning tasks, an RDF inference layer providing rule reasoning support and an OWL-DL reasoner. Besides providing reasoning support, they are responsible for collecting the status of SDOs published using the OWL format, and for building an integrated OWL description accessible to reasoning.

The main feature that distinguishes the latter study is the lack of a central reasoning or control entity; it is fully decentralized. Collecting the reasoning tasks in a central entity certainly has many advantages; we can achieve better control, and better coordination between the various entities that have access to the central entity. Blackboard-based and shared-memory models have been thoroughly studied and used in many different types of distributed systems and have proved to work well in practice. The requirements are, though, much different in this setting. Context may not be restricted to a small room, office or apartment; we must also study cases of broader areas. The communication with a central entity is not guaranteed; we must assume unreliable and restricted wireless communications. Thus, a fully distributed scheme is a necessity. The OWL-SF framework is a step towards the right direction, but certainly not the last one. In order to deal with more realistic ambient environments, we need to eliminate some of the assumptions that they make. For example, different entities are not required to use the same representation and reasoning models, and we cannot always assume the existence of dedicated reasoning machines.

5 Other Reasoning Techniques

This section presents additional techniques that have been used to enhance the reasoning capabilities of AmI applications to deal with certain challenges, such as the ambiguity of context information, and the vast amount of context data.

In AmbieSense ([27]), they deal with the potentially vast amount of context data, using *Case Based Reasoning*. The reasoning mechanism is split into two different parts; the on-line part that resides on the user's mobile device, and the off-line part that resides on the user's backbone system. When new information arrives from the context retrieval module, it is translated to fit a preexistent ontology and sent to a CBR agent. The agent tries to retrieve a known context or case, and classifies the current situation based on the retrieved one. The associated goal is then presented to the task decomposition agent, and the case is stored in the case base. Since the user is expected to experience a few different situations daily, the storage of the cases will quickly fill up the mobile device and the CBR searching process will be hampered. To remedy this, some of the reasoning process is moved into the user's backbone servers.

The ec(h)o audio museum guide, described in [28], uses DAML+OIL ontologies for the representation of context data and user profiles. Its reasoning engine uses a forward-chaining reasoning mechanism to select the sound objects to be presented. The rules use several criteria that correspond to the semantic descriptions of the museum artifacts, the visitor's profile, and the way the visitor moves and interacts with the artifacts. To perform reasoning more efficiently, they build a virtual network that keeps track of possible combinations of facts, and support rule activation using the RETE algorithm (implemented in Jess).

The use of a Bayesian network to deal with the ambiguity of context data has been proposed in some recent studies. In MIRA, a context-based retrieval system capable of recording and indexing MBone videoconferences, they use a Bayesian network, coupled to a cost model, to describe a context-retrieval service that provides performance measures based on reliability and resource usage cost ([29]). In [30], a probabilistic model is used to define uncertain contexts. This model extends the OWL ontology model of SOCAM, by attaching probability values to the context predicates. They also adopt a Bayesian network as an underlying reasoning mechanism, as it has efficient probabilistic reasoning capabilities and allows representing causal relationships between various contexts. Bayesian networks to recognize high-level contexts have also been used in [25].[1]

6 Discussion

The special requirements of ambient environments impose the need of logic models that inherently deal with the imperfect nature of context data. Models that embody the notions of uncertainty, temporal and spatial change, and incompleteness would provide more robust and efficient solutions. A possible solution is the use of *nonmonotonic* reasoning, which has already been studied and used in other settings with similar requirements, such as the Web, e-learning environments, business rules, security specifications, negotiation protocols, and others. Recently, a number of nonmonotonic rule languages have been studied and reasoners that integrate them well with ontologies have been developed.

The main drawback of this approach is its relatively higher computational complexity, which becomes even worse, if we consider the potentially vast amount of available context data. A possible solution is to partition the large knowledge bases into smaller pieces, share these pieces with other computing devices, and deploy some form of partition-based reasoning. This is of course not an easy task, and only few recent studies have focused on this problem. An interesting approach is proposed in [31], which studies the partitioning of a large OWL ABox with respect to a TBox so that specific kinds of reasoning can be performed separately on each partition and the results trivially combined in order to achieve complete answers. In [32], they propose algorithms for reasoning with partitions of related logical axioms in propositional and first-order logic, and a greedy algorithm that automatically decomposes a set of logical axioms into partitions. Applying these ideas in AmI seems to be a very promising research direction.

Finally, to achieve collective intelligence, we must study methods for integrating and reasoning with data coming from heterogeneous sources and possibly described in different vocabularies. Translating all the data in a common format (schema) and performing centralized reasoning (followed by most of the studies that we presented) is one of some possible solutions. This approach is described as the Local-As-View approach in the Data Integration research area ([33]). Other approaches, concerning mainly the integration of heterogeneous data, are the Global-As-View approach and the Both-As-View approach ([33]), which have been recently studied and implemented in semantic P2P management systems. GAV assumes a global virtual schema, which is defined as a set of views over the data source schemas. This enables writing queries and rules using the local language of each data source. In BAV, local schemas are mapped to

[1] The modeling and reasoning approaches, along with the architecture and the aim of the systems referenced in Sections 2-6 are summarized in Table 1.

Table 1. Main Features of Context-Aware Frameworks

System	Modeling	Reasoning	Architecture	Aim
CoBrA [9]	OWL	RDQL	centralized (agent-based)	context-aware services
Context Awareness Framework [10]	RDF	RDQL	centralized	service prioritization
CG Platform [11]	RDF	RQL	centralized	location-based services
Semantic Mobile Environment [13]	DL	DL	distributed (P2P)	profile-service matchmaking
Context-Aware Door Lock [14]	OWL	DL	centralized	automatic door lock
SOCAM [16],[30]	OWL	FOL + Bayesian	centralized (middleware)	middleware for mobile services
Semantic Space [17]	OWL	RDQL+FOL	centralized	smart space mobile services
Gaia Context Infrastructure [19]	FOL	FOL	centralized	context-aware services
CONON Prototype [18]	OWL	DL+FOL	centralized	context-aware services
eWallet [20]	OWL	Jess	centralized (agent-based)	context-aware services
Context-Aware Access Control Framework [21]	OWL	DL+LP	centralized	policy evaluation
CARE [22]	OWL	DL+LP	centralized (middleware)	service adaptation
sTuples [23]	DAML+OIL	DL	decentralized shared memory	mobile services
Semantic Spaces [24]	RDF		decentralized shared memory	information sharing
Context Management Framework [25]	RDF	Bayesian	decentralized (blackboard-based)	information sharing notification services
OWL-SF [26]	OWL	DL	distributed (SDOs)	distributed services
AmbieSense [27]	taxonomies	CBR	centralized	context management
ec(h)o system [28]	DAML+OIL	Jess	centralized	audio museum guide
MIRA [29]	XML	Bayesian	centralized	videoconferences management

each other using a sequence of schema transformations (*mappings*). Reasoning with multiple ontologies interrelated with semantic mappings is studied in [34]. Examples of totally distributed reasoning algorithms, where the whole reasoning procedure can be viewed as a chain of reasoning tasks performed by different entities, can be found in [35]. These approaches can also lead to new ideas on how to exploit the different reasoning capabilities of each entity in an ambient environment, in order to make the whole system of entities more intelligent.

References

1. Henricksen, K., Indulska, J.: Modelling and Using Imperfect Context Information. In: Proceedings of PERCOMW '04, Washington, DC, USA, IEEE Computer Society (2004) 33–37
2. Schmidt, A.: Interactive Context-Aware Systems Interacting with Ambient Intelligence. In Riva, G., Vatalaro, F., Davide, F., Alcaniz, M., eds.: Ambient Intelligence, IOS Press (2005)
3. Seaborne, A.: RDQL - A Query Language for RDF. W3C member submission, Hewlett Packard (2004)
4. Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., Scholl, M.: RQL: a declarative query language for RDF. In: Proceedings of the 11th International World Wide Web Conference (WWW). (2002) 592–603
5. Sintek, M., Decker, S.: TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web. In: Proceedings of the First International Semantic Web Conference, Springer-Verlag (2002) 364–378
6. FaCT: FaCT system website. http://www.cs.man.ac.uk/ horrocks/FaCT/ (2003)
7. Haarslev, V., Möller, R.: RACER System Description. In: IJCAR. (2001) 701–706
8. Parsia, B., Sirin, E.: Pellet: An OWL DL Reasoner. In: 3rd International Semantic Web Conference (ISWC2004). (2004)
9. Chen, H., Finin, T., Joshi, A.: Semantic Web in a Pervasive Context-Aware Architecture. Artificial Intelligence in Mobile System 2003 (2003) 33–40
10. Forstadius, J., Lassila, O., Seppanen, T.: RDF-based model for context-aware reasoning in rich service environment. In: PerCom 2005 Workshops. (2005) 15–19
11. Patkos, T., Bikakis, A., Antoniou, G., Plexousakis, D., Papadopouli, M.: A Semantics-based Framework for Context-Aware Services: Lessons Learned and Challenges. In: Proceedings of 4th International Conference on Ubiquitous Intelligence and Computing (UIC-2007). (2007) accepted for publication
12. van Bunningen, A.: Context aware querying - Challenges for data management in ambient intelligence. Technical Report TR-CTIT-04-51, Univ. of Twente (2004)
13. von Hessling, A., Kleemann, T., Sinner, A.: Semantic User Profiles and their Applications in a Mobile Environment. Fachberichte Informatik 2–2005, Universität Koblenz-Landau (2005)
14. Turhan, A.Y., Springer, T., Berger, M.: Pushing Doors for Modeling Contexts with OWL DL a Case Study. In: PERCOMW '06, Washington, DC, USA, IEEE Computer Society (2006)
15. Haarslev, V., Möller, R., Wessel, M.: RacerPro User Guide. http://www.racer-systems.com/products/racerpro/users-guide-1-9.pdf (2005)
16. Gu, T., Pung, H.K., Zhang, D.Q.: A Middleware for Building Context-Aware Mobile Services. In: Proceedings of the IEEE Vehicular Technology Conference (VTC 2004), Milan, Italy (2004)

17. Wang, X.H., Dong, J.S., Chin, C.Y., Hettiarachchi, S.R., Zhang, D.: Semantic Space: an infrastructure for smart spaces. IEEE Pervasive Computing **3**(3) (2004) 32–39
18. Wang, X.H., Zhang, D.Q., Gu, T., Pung, H.K.: Ontology Based Context Modeling and Reasoning using OWL. In: PERCOMW '04, Washington, DC, USA, IEEE Computer Society (2004) 18
19. Ranganathan, A., Campbell, R.H.: An infrastructure for context-awareness based on first order logic. Personal Ubiquitous Comput. **7**(6) (2003) 353–364
20. Gandon, F.L., Sadeh, N.M.: Semantic web technologies to reconcile privacy and context awareness. Journal of Web Semantics **1** (2004) 241–260
21. Toninelli, A., Montanari, R., Kagal, L., Lassila, O.: A Semantic Context-Aware Access Control Framework for Secure Collaborations in Pervasive Computing Environments. In: Proc. of 5th International Semantic Web Conference. (2006) 5–9
22. Agostini, A., Bettini, C., Riboni, D.: Experience Report: Ontological Reasoning for Context-aware Internet Services. In: PERCOMW '06, Washington, DC, USA, IEEE Computer Society (2006)
23. Khushraj, D., Lassila, O., Finin, T.: sTuples: Semantic Tuple Spaces. In: First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous04). (2004) 267–277
24. Krummenacher, R., Kopecký, J., Strang, T.: Sharing Context Information in Semantic Spaces. In: OTM Workshops. (2005) 229–232
25. Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H., Malm, E.J.: Managing Context Information in Mobile Devices. IEEE Pervasive Computing **02**(3) (2003) 42–51
26. Mrohs, B., Luther, M., Vaidya, R., Wagner, M., Steglich, S., Kellerer, W., Arbanowski, S.: OWL-SF - A Distributed Semantic Service Framework. In: Workshop on Context Awareness for Proactive Systems (CAPS 2005). (2005)
27. Kofod-Petersen, A., Mikalsen, M.: Representing and Reasoning about Context in a Mobile Environment. Revue d'Intelligence Artificielle **19**(3) (2005) 479–498
28. Hatala, M., Wakkary, R., Kalantari, L.: Ontologies and rules in support of real-time ubiquitous application. Journal of Web Semantics, Special Issue on "Rules and ontologies for Semantic Web" **3**(1) (2005) 5–22
29. Castro, P., Mani, M., Mathur, S., Muntz, R.R.: Managing Context for Internet Videoconferences: The Multimedia Internet Recorder and Archive. In: Proc. of Multimedia and Computer Networks, San Jose, California (2000)
30. Gu, T., Pung, H.K., Zhang, D.Q.: A Bayesian Approach for Dealing with Uncertain Contexts. In: Proceedings of the Second International Conference on Pervasive Computing, Vienna, Austria, Austrian Computing Society (2004)
31. Guo, Y., Heflin, J.: A Scalable Approach for Partitioning OWL Knowledge Bases. In: Second International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2006). (2006)
32. Amir, E., McIlraith, S.: Partition-based logical reasoning for first-order and propositional theories. Artif. Intelligence **162**(1-2) (2005) 49–88
33. Halevy, A.Y., Rajaraman, A., Ordille, J.J.: Data Integration: The Teenage Years. In: VLDB. (2006) 9–16
34. Serafini, L., Tamilin, A.: DRAGO: Distributed Reasoning Architecture for the Semantic Web. In: ESWC. (2005) 361–376
35. Adjiman, P., Chatalic, P., Goasdoue', F., Rousset, M.C., Simon, L.: Distributed Reasoning in a Peer-to-Peer Setting: Application to the Semantic Web. Journal of Artificial Intelligence Research **25** (2006) 269–314