AUTOMATIC CONFIGURATION OF SERVICE DIRECTORIES IN MULTIAGENT SYSTEMS

Martin Krebs, Karl-Heinz Krempels, Janno von Stülpnagel and Christoph Terwelp
Informatik 4, Intelligent Distributed Systems Group, RWTH Aachen University, Aachen, Germany

Keywords: Distributed Service Directory, Automatic Configuration, mDNS, DNS-SD, Multiagent System.

Abstract: Service directories provide basic functionalities for service discovery and service announcement in Multiagent Systems (MAS). A manual configuration of distributed MAS and their service directories is a time consuming and complex task and thus an automatic mechanism is highly desirable. Furthermore, the requirement for a fast response and processing time for agent queries still remains. Existing approaches do not fulfill these requirements in a satisfactory way. Thus, a new approach based on multicast DNS and DNS Service Discovery (DNS-SD) was developed and implemented for the multiagent framework JADE (Java Agent DEvelopment Framework). The architecture enables the automatic configuration of agent platforms in dynamic local networks as well as the later federation of the service directories which is necessary for a distributed service discovery operation.

1 INTRODUCTION

Service directories (SD) contain service descriptions of services provided by agents and agent plattforms. Agent interactions are based on service requests or queries among agents. To determine the participants for an interaction an agent has to query a directory for suiteable agents. Thus, service directories provide essential support for agent interactions. SD are provided on Agent Platforms (AP) as a basic service for agents. Every agent that provides a service for other agents has to register this service with the service directory. Therefore, the agent submits to the service direcory a formal service description that contains the service name, the ID of the service providing agent, ontologies supported by the service, and interaction protocols supported by the agent to access the service.

The paper is outlined as follows: Section 2 gives an overview of existing solutions for service directories in distributed MAS and their drawbacks. In section 3 a new approach for a distributed service directory is introduced. The implementation of the approach is discussed in section 4 and its evaluation in section 5. Conclusions and future developements are discussed in section 6.

2 STATE OF THE ART

Service directories provide basic functionalities for service discovery and therewith for the interaction among agents. For this reason service directories are subject of standards related to agent technology. The *Directory Facilitator* (DF) specifiaction (FIPA-SC00001G, 2002) describes the interface of the service directory, the format for agent and service descriptions, and the ontologies and interaction protocols supported by the DF. Existing implementations of the DF specification use *polling* oder *lease-time* mechanisms to keep the registered service description up to date:

Polling: consists of periodic update requests initiated by the service directory for registered service descriptions, independent of the rate of change of the information. The time interval for update requests is constant for polling.

Lease-time: expects information updates from each registered service by itself. For each service a time interval is defined in which the next update has to take place. Otherwise the service is removed from the directory.

Krebs M., Krempels K., von Stülpnagel J. and Terwelp C. (2009).
AUTOMATIC CONFIGURATION OF SERVICE DIRECTORIES IN MULTIAGENT SYSTEMS.
In *Proceedings of the International Conference on Agents and Artificial Intelligence*, pages 558-563
DOI: 10.5220/0001791105580563

The first world wide service directory infrastructure for MAS was implemented in the *Agentcities* project (Willmott, 2004), founded by the EU. In the Agentcities service directory the registered agent and service descriptions were kept up to date with the help of the polling mechanism. During the essential operation time the number of registered services touched 200 agent platforms and about 1000 agents. The minimal information update cycle caused by this polling load was close to 5 minutes. This interval was however to long for dynamic agent based applications with short-lived agents (Kirn et al., 2003) (Kirn et al., 2006) (Woelk et al., 2006).

A new approach for distributed service directories was developed in the *X-OpenNet* project (Willmott et al., 2004a), the successor of the Agentcities project. This approach was based on a two-level architecture. The upper level consisted of a full-meshed federation of 16 nodes, which were monitored by all other nodes of the same level with the help of a combined *polling-lease-time*-mechanism. The lower level of the architecture consisted of the local DF of each connected agent platform. This DF either answered a search request itself or forwarded it to one of two preselected nodes of the upper layer. The complex configuration task in the upper layer for the deployment of *X-OpenNet* infrastructure services caused a low user acceptance and popularity of the whole infrastructure.

In the following years many approaches from the *peer to peer* (P2P) area have been considered to remove the drawbacks from the upper level of the *X-OpenNet* architecture. Particularly the creation of federations with the help of the *JXTA* Framework (Brookshier et al., 2002) (FIPA-PC00096, 2004) (Chen et al., 2006) and the *Jabber* Framework (Cámara et al., 2006) (Fischer et al., 2006).

The *JXTA* Framework provides a robust infrastructure for P2P networks. It consists mainly of service directories, signaling mechanisms, and programming interfaces for the development of client applications. In (Chen et al., 2006) (FIPA-PC00096, 2004) the integration of the *JXTA*-Framework with JADE is discussed. The paper states that the deployment effort of an distributed agent platform can be reduced by the configuration of the service directory, but is still bound by the configuration time of the agent platform. Furthermore, the signalling load produced by the JXTA framework seems to be very high. Thus, this approach seems to remove only a part of the drawbacks.

The *Jabber* Framework consists of a collection of network protocols for instant messaging, file transfer and signalling to transport messages and files, and to show the presence of communication parties. The integration of the *Jabber* Framework with JADE is simpler to implement than the integration of the *JXTA*-Framework. Furthermore, the setup of a *Jabber*-client is much easier than the setup of a *JXTA*-client. However, the drawbacks of this integration include also the signalling load and the requirement that all the communication parties have to be well known and have to be provided in advance. The *Jabber* Framework supports the discovery of communication parties only with the help of a centralized directory.

All approaches presented here are a tradeoff between the required configuration effort for the communication interface of an agent platform, the limitations of the actuality of registration records in the service directories, and the distribution level of a MAS.

The configuration of service directories in distributed MAS is a complicated and time-consuming job that becomes more complex in dynamic network environments, like ad-hoc networks. However, this setup is often provided for the demonstration track in conferences and in developer workshops in academic or touristic environments.

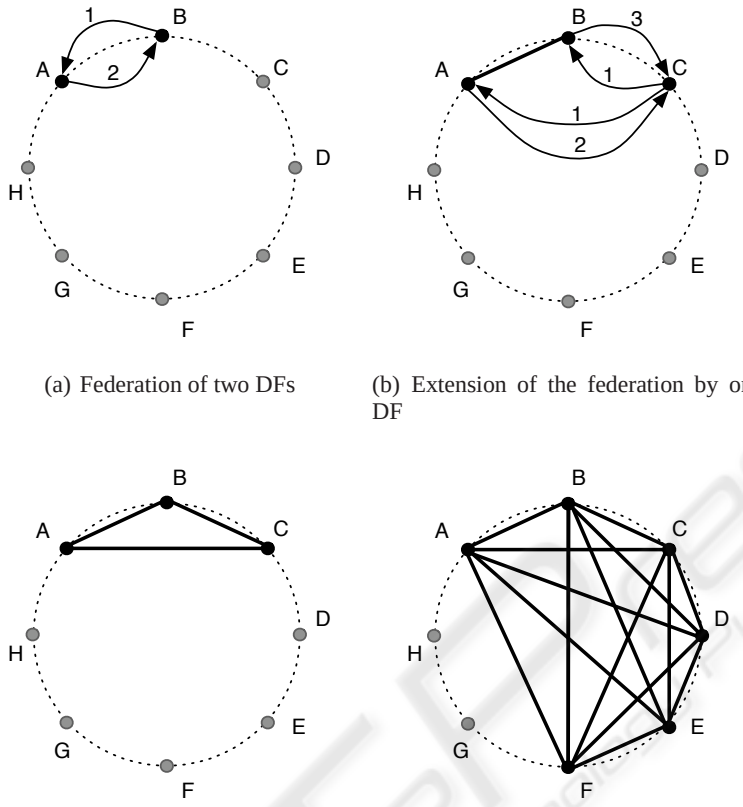
Thus, to become able to show distributed prototypes of agent-based applications at a conference, e.g. (Kirn et al., 2003) (Kirn et al., 2006) (Woelk et al., 2006), or to setup a development environment for agent-based applications, e.g. (Dale et al., 2003) (Willmott, 2004) (Willmott et al., 2004b) (Scholz et al., 2005), there is a need to configure distributed MAS and to federate their service directories in a fast way.

3 APPROACH

The presented architecture for automatic configuration of service directories in MAS is based on a flat structure, i.e. there is no hierarchy. All agent platforms which provide a directory service establish a fully-meshed federation. This meshed graph is the basis for an efficient distributed service discovery operation. The corresponding agent platforms are configured automatically with a configuration and signaling system which ensures that all nodes use the same IP address range and namespace.

In the following, we first present the setup of the directory federation in an ad-hoc network. In the next step the processing of the search queries in the federation is explained in detail which is followed by a description of the used signaling and configuration system.

Every node which offers a directory service announces its availability in the network, see Fig. 1. A



(a) Federation of two DFs

(b) Extension of the federation by one DF

(c) Federation with three DFs

(d) Federation with six DFs

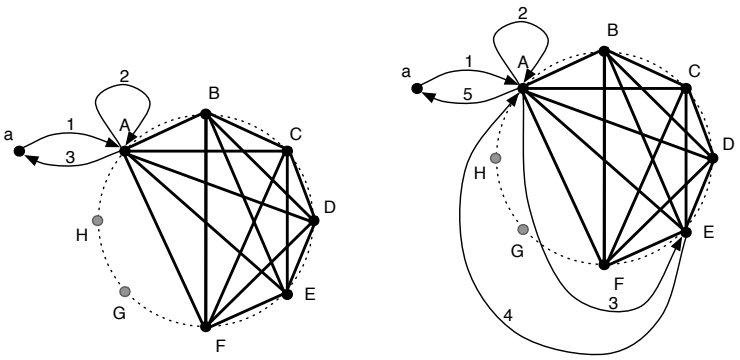
Figure 1: Federation of Service Directories.

federation is established if there are at least two nodes which intend to form a federation. The Figs. 1(a) up to 1(d) show the incremental expansion with the nodes *C*, *D*, *E*, and *F*. Finally, the federation forms a fully-meshed graph. The main advantage of a flat directory structure is the high actuality of the resource records, because the registration, update and delete operation is done at the corresponding local agent platform. Accordingly, most of the queries can be replied directly by this platform which is based on the assumption that an agent interacts most of the time with other agents of its direct neighborhood or otherwise prefers a migration (Diepolder and Krempels, 2006) (Krempels et al., 2006).

The discovery of a service in the established federation is depicted in Fig. 2. The most common case is shown in Fig. 2(a) where a search for a service can be satisfied by the local AP. The request (1) of agent *a* at directory *A* is realized by an internal lookup at directory (2). Finally, the matching services are sent back to the requesting agent (3). If the service request does not match at the local directory (see Fig. 2(b)), the request is forwarded (3) to the directory *E* and replied by it (4). The reply is finally forwarded to the requesting agent. The most complex case occurs (see Fig. 2(c)), if the request of agent *a* cannot be replied by the local directory (2) or the update of an outdated resource record fails (3,4). In this case it is required, that the request is forwarded to all directories in the federation (5) and only the successful replies are sent back to the local directory *A* (6).

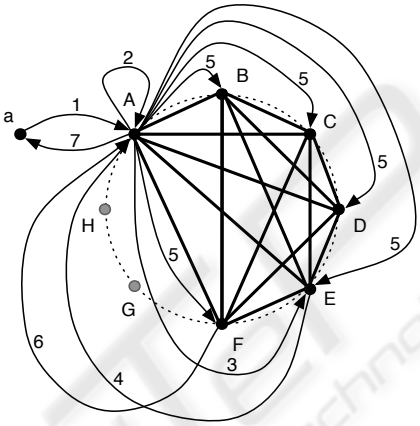
In environments with dynamically assigned addresses und network names for agent platforms it is necessary, that agent platforms are configured automatically. Therefore, the protocols mDNS (Multicast Domain Name System) and DNS-SD (Domain Name System based Service Discovery) were chosen (Cheshire and Krochmal, 2005) (Steinberg and Cheshire, 2005). This approach satisfies the following requirements:

- automatic assignment of IP-addresses without a central DHCP server;
- automatic DNS name resolution and name assignment without DNS or DHCP server;



(a) Answer of the search request from th- elocal DF

(b) Answer of the search request from a (selected) DF of the federation



(c) Recursive answers of the search request by the federated DFs

Figure 2: Processing of a search request by federated service directories.

- service discovery and registration in a local network.

The here presented approach for the automatic configuration of directories in MAS allows for efficient administration of service information. Furthermore, the fully-meshed structure allows an efficient distributed search for services. The configuration and federation of the architecture is done automatically with mDNS and DNS-SD.

4 ARCHITECTURE

The approach was implemented for the Agent-Framework JADE (Bellifemine et al., 2007). The JmDNS Library has been used as mDNS and DNS-SD protocol implementation. The work done here was focussed on the integration of JmDNS into JADE and the implementation of a configuration service for the service directories. JADE provides most basic services as so-called *Kernel-Services*. JmDNS was integrated as such a Kernel-Service. Every agent only accesses the local service directory as it is done in the non-distributed case. The directory services access JmDNS through the Kernel-Service. As shown in Fig. 4 when a directory service starts or stops it broadcasts its state change through the *serviceAdded* and *serviceRemoved* methods to the other agent platforms in the local network. Such anouncements from other agent platforms are processed by the *DFServiceListener* and a federation is build.

5 EVALUATION

The implementation of this approach was evaluated in a setup with four computers interconnected in a WiFi ad-hoc network. The WiFi network interfaces

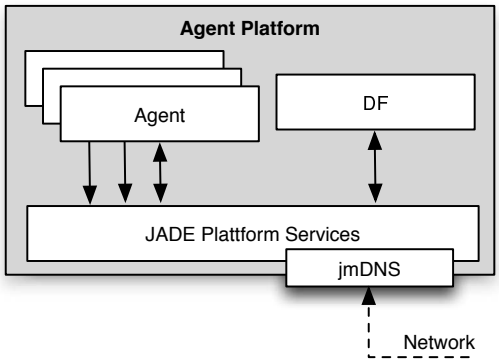


Figure 3: Integration of JmDNS in JADE.

of these computers were configured by the mDNS / DNS-SD implementations of their corresponding operating system: e.g avahi[1] on Linux, Bonjour on MacOS X[2] and Windows XP[3]. After the completion of the network interface configuration process an agent platform was launched on each computer to serve as measurement platform. For all measurements made the federation of the service directories was established immediatly after the launch of the platform.

On three of this platforms an agent was launched that registered 2000 services in the service directory. On the fourth platform a search agent was started that requested the local service directory to search a service with exactly one match on a service description on each of the other platforms.

Table 1: Search time in four platforms.

measurement	search time (ms)
1	94
2	62
3	63
4	78
5	63
6	62
7	63
8	62
9	78
10	94
average	71,9

For the first series of measurements shown in Tab. 1 the platforms were restarted for each measurement. The resulting mean value of 71,9 ms for the distributed search processing is sufficiently fast to be used in dynamic agent based scenarios, e. g. the ones discussed in (Kirn et al., 2003) (Kirn et al., 2006) (Woelk et al., 2006).

[1]http://avahi.org

[2]Bonjour and MacOS X are registered trade marks of Apple Inc.

[3]Windows XP is a registered trade mark of Microsoft Inc.

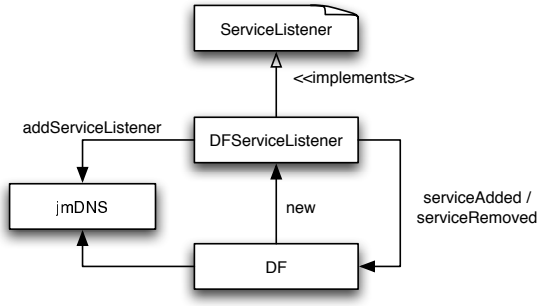


Figure 4: Kernel-Service for *JmDNS*.

6 CONCLUSIONS AND OUTLOOK

The combination of the agent framework JADE and the JmDNS implementation of the mDNS / DNS-SD protocols is an efficient approach for easy, fast and mostly automated configuration of distributed MAS in local networks. Even for large numbers of registered services a fast distributed search is possible.

The discussed improvements of the JADE framework for automated configuration and federation of service directories were presented to the JADE community and will be included in the next release of the JADE framework.

ACKNOWLEDGEMENTS

This work is motivated by experiences made in the German priority research programme 1083 Intelligent Agents in Real-World Business Applications (2000-2008).

Contributions to the implementation of the presented approach and to its evaluation were made by Milan Fort, Sven Lilienthal, Andreas Kelle-Emden, and Moaffak Assassa. Special thanks to Uta Christoph for her critical suggestions, lovely reviews, and unlimited patience.

REFERENCES

Bellifemine, F. L., Caire, G., and Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*. Wiley.

Brookshier, D., Govoni, D., Krishnan, N., and Soto, J. C. (2002). *JXTA: Java P2P Programming*. Sams, Indianapolis, IN, USA.

Cámara, J. P., Gregori, M. E., Bada, G. A., García-Fornes, A., Julián, V., and Botti, V. J. (2006). Adding new communication services to the fipa message transport system. In (Fischer et al., 2006), pages 1–11.

Chen, E., Sabaz, D., and Gruver, W. (2006). Jade and jxta extensions for the implementation of distributed systems. *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on*, 1:740–745.

Cheshire, S. and Krochmal, M. (2005). DNS-Based Service Discovery. Technical report, Internet Engineering Task Force.

Dale, J., Burg, B., and Willmott, S. (2003). The agentcities initiative: Connecting agents across the world. *Innovative Concepts for Agent-Based Systems*, pages 453–457.

Diepolder, S. and Krempels, K. H. (2006). Mobile agenten. mobile agents. *PIK - Praxis der Informationsverarbeitung und Kommunikation*, 29(4).

FIPA-PC00096 (2004). FIPA JXTA Discovery Middleware Specification.

FIPA-SC00001G (2002). FIPA Abstract Architecture Specification.

Fischer, K., Timm, I. J., André, E., and Zhong, N., editors (2006). *Multiagent System Technologies, 4th German Conference, MATES 2006, Erfurt, Germany, September 19-20, 2006, Proceedings*, volume 4196 of *Lecture Notes in Computer Science*. Springer.

Kirn, S., Heine, C., Herrler, R., and Krempels, K.-H. (2003). Agent.Hospital - Agentenbasiertes offenes Framework für klinische Anwendungen. In Uhr, W., Esswein, W., and E.Schoop, editors, *Medien - Märkte- Mobilität*, pages 837–857. Wirtschaftsinformatik 2003, Physica Verlag, Heidelberg.

Kirn, S., Herzog, O., Lockemann, P., and Spaniol, O. (2006). *Multiagent Engineering: Theory and Applications in Enterprises*. International Handbooks on Information Systems. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Krempels, K.-H., Spaniol, O., Scholz, T., Timm, I., and Herzog, O. (2006). Interaction design. *Multiagent Engineering*, pages 383–403.

Scholz, T., Krempels, K.-H., Nimis, J., Schiemann, B., Woelk, P.-O., Braubach, L., and Pokahr, A. (2005). www.agententerprise.net – a mmas-based web-portal for planning and Control of Complex Manufacturing Supply Chains managed by ASCML – the Agent Society Configurator Manager and Launcher. In *AAMAS 2005 Demonstration Proceedings*, volume openNet Networked Agents Demonstration for AAMAS 2005.

Steinberg, D. H. and Cheshire, S. (2005). *Zero Configuration Networking*. O'Reilly. Not yet released.

Willmott, S. (2004). Deploying intelligent systems on a global scale. *Intelligent Systems, IEEE*, 19(5):71–73.

Willmott, S., Bonnefoy, D., Constantinescu, I., Thompson, S., Charlton, P., Dale, J., and Zhang, T. (2004a). Agent based dynamic service synthesis in large-scale open environments: experiences from the agentcities testbed. *Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004. Proceedings of the Third International Joint Conference on*, pages 1318–1319.

Willmott, S., Rana, O., Krempels, K.-H., McBurney, P., and Weichart, G. (2004b). Networked Agents: Towards Large-scale Deployment of Agents in Open Networked Environments (NET AGENTS). *AgenLink News*, (16):22–23.

Woelk, P.-O., Rudzio, H., Zimmermann, R., and Nimis, J. (2006). Agent.enterprise in a nutshell. *Multiagent Engineering*, pages 73–90.