A Dynamic Model to enhance the Distributed Discovery of services in P2P Overlay Networks

Adel Boukhadra, Karima Benatchba, and Amar Balla

Ecole national Supérieur d'Informatique, BP 68M, Oued-Smar, 16309, Alger, Algérie
a_boukhadra@esi.dz, k_benatchba@esi.dz, a_balla@esi.dz

Abstract. In Service Computing (SC), online Semantic Web services (SWs) is evolving over time and the increasing number of SWs with the same function on the Internet, a great amount of candidate services emerge. So, efficiency and effectiveness has become a stern challenge for distributed discovery to tackle uniformed behavior evolution of service and maintain high efficiency for large-scale computing. The distributed discovery of SWs according to their functionality increases the capability of an application to fulfill their own goals. In this paper, we describe an efficient and an effective approach for improving the performance and effectiveness of distributed discovery of SWs in P2P systems. As most Web services lack a rich semantic description, we extend the distributed discovery process by exploiting collaborative ranking to estimate the similarity of a SWs being used by existing hybrid matching technique of OWL-S (Ontology Web Language for Services) process models in order to reduce costs and execution time. We mapped our distributed discovery of OWL-S process models by developing a real application based on Gamma Distribution; a technique used to decrease the bandwidth consumption and to enhance the scalability of P2P systems. The particularity of the Gamma Distribution is then integrated for disseminating request about the P2P networks to perform quality based ranking so that the best SWs can be recommended first. The experimental result indicates that our approach is efficient and able to reduce considerably the execution time and the number of message overhead, while preserving high levels of the distributed discovery of SWs on large-size P2P networks.

Keywords: SWs, Distributed Discovery, P2P Computing, Gamma Distribution, Matching of Ontology, OWL-S process model.

1 Introduction

Service-oriented computing (SOC) has emerged as an effective means of developing distributed applications and an accurate assessment the full potential of reputation which is essential for discovering between alternative Web services. Discovery of Web service is a great interest and is a fundamental area of research in distributed computing [6].

C. Badica et al. (eds.), *Intelligent Distributed Computing X*,
Studies in Computational Intelligence 678, DOI 10.1007/978-3-319-48829-5_2

Actually, with the increasing number of SWs with the same function, in an open and dynamic environment, such as Internet, a great amount of candidate services emerge. Moreover, the number of registries that offer available Web services is also increasing significantly. It is not always easy to find services that matching users queries. Find a service from the candidate Web service set, according to the requirements of end-users has become a key hindrance and a strenuous task even for an experienced user. The Web service discovery satisfying the query is led by an optimization process, aiming to achieve the best SWs in the end [1] [4] [5] [7].

This could be very challenging to end-users given the huge number of available Web services online, who have to decide where to fulfill their requests: on local directory or to a replace this structure by several registries properly organized to support the dynamic, flexible and more efficient request propagation in a highly-dynamic distributed computing. Due to low accuracy, poor performance, high operational and maintenance cost, and sometimes for low availability of the functionality of SWs, the distributed discovery of SWs on the P2P systems provides a unique opportunity to address the above challenges [1] [7]. In recent years, the rise of P2P networks is attested by the increasing amount of interest in both commercial and academic areas to develop systems for data sharing simple and effective with many advantages such as decentralization, self-organization, autonomy, etc. At the same time, the SWs community has been slowly evolving toward a higher degree of distribution [4] [5] [6].

In this paper, we propose an efficient and an effective approach that addresses some aspects related to problems the time complexity of collaboration in the process of automatic discovery for SWs in P2P computing. For this purpose, our approach is based on P2P computing that proved to be scalable, more fault tolerant by eliminating the single point of failure, efficient by reducing the overhead of centralized update of the service discovery and robust solutions for distributed discovery of SWs. Specifically, our contributions in this work are summarized as follows. We exploit hybrid matching technique of OWL-S process models to develop functional features of Web services and the desired specification given by the user. Both the Web services and request can be then represented as OWL-S process model. We incorporate Gamma Distribution based collaborative ranking to identify additional functionally relevant Web services, in order to efficiently and effectively discover appropriate Web services distributed among all peers in a large-size P2P network. The idea of using Gamma Distribution is to manage large and continuously growing spaces of Web services with reasonable resolution times. The top-k most possible Web services are discovered, which are all considered as functionally relevant to the new Web services.

The remainder of this work is organized as follows: Section 2 describes the related work. In Section 3 we focus on the proposed distributed approach for discovering SWs in the unstructured P2P computing. An experimental evaluation is presented in Section 4, and, finally, in Section 5, we present our conclusion and highlight our future work directions.

2 Related Work

In this section, we discuss several representative related work and differentiate them with our work.

In [2], authors propose the P2P-based Semantic Driven Service Discovery (P2P-SDSD) framework to enable cooperation and communication based on a semantic overlay that organizes semantically the P2P-integrated knowledge space and emerges from local interactions between peers. The semantic overlay can be seen as a continuously evolving conceptual map across collaborative peers that provide similar services and constitute synergic service centers in a given domain. The semantic overlay enables effective similarity based service search and optimization strategies are defined for request propagation over the unstructured P2P network keeping low the generated network overload. Each collaborative peer in the unstructured P2P network has a local knowledge infrastructure constituted by: (i) UDDI Registry; (ii) Peer Ontological Knowledge, that provides a conceptualization of abstract service operations and Input/output parameters through a given domain ontology; a conceptualization of service categories through a Service Category Taxonomy (SCT).

In [7], present a technique to improve discovery in unstructured P2P service networks, based on a probabilistic forwarding algorithm driven by network knowledge, such as network density, and traces of already discovered service compositions (CONs). The technique aims at reducing the composition time and the messages exchanged during composition, relying on two considerations: if the network is dense, forwarding can be limited to a small number of neighbors; if the network is semi structured in CONs, forwarding can be directed to the super peers that may own the desired information. The approach improves the discovery and composition process by using distributed bidirectional search. The benefit is twofold: first, it is possible to have concurrent searches in a P2P service network in both goal directions (from pre- to post- and from post to preconditions), reducing the response time when solutions are present; second, when no complete solution for a goal is present, gaps in partial found solutions can be identified. This way, it is possible to have feedbacks about users' most required unavailable business operations, allowing providers to discover new business opportunities.

In [11], authors present a distributed approach to SWs publication and discovery by leveraging structured P2P network. In this work, the computers concerned constitute a P2P network to maintain the sharable domain and service ontologies to facilitate SWs discovery. When a requestor submits a semantic query for desired services, the P2P network can effectively obtain semantically qualified services. The main contributions of this work can be summarized as follows: this approach introduces a semantic-based service matching rule. In order to achieve the optimal match between a query, it proposes a concept of Ordered-Concept-Tree (OCT) to semantically sort the relevant concepts for service matching. In addition, to freely share and make full use of the semantic concepts defined in ontologies for OCT construction, it also proposes a method to publish ontologies to structured P2P network.

3 The Proposed Approach

In this section, we describe in detail the proposed approach that uses dynamic topology adaptation to improve the efficiency and the effectiveness of distributed discovery of SWs.

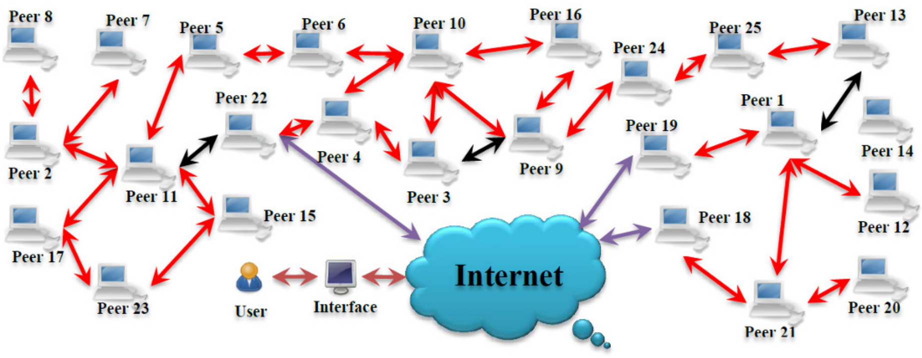

Fig. 1. Unstructured P2P Approach for Discovering SWs.

We propose further model which defines the topology of the P2P computing for supporting communication and collaboration between different Web service by the means of an effective use of the resources of P2P network and the management of scalability in an integrated and practical way. Such our model can support scalability, maximize search recall, reduce the query traffic and must also be able to achieve an acceptable performance. There are mainly three different architectures for P2P systems: hybrid, pure and unstructured; but the unstructured P2P systems are most commonly used in today's Internet [5] [6] [11].

In unstructured P2P network, we distinguish between two types of peers, neighbor and ultra-peer. Each peer is linked to a set of other peers in the P2P network via bi-directional connections, that is, each peer has a limited number of neighbors and it can choose the appropriate other neighbors it wishes to connect when necessary. When a peer receives a search request, it manages the discovery process locally and/or it will forward the query from other neighbors in the P2P network using the dynamic querying algorithm.

If a peer frequently returns good results of a large number of queries from one of its neighbors, it is likely to have common collaboration with the requesting peer to discover the appropriate Web services. It will mark this neighbor (the requesting peer) as an ultra-peer of the requesting peer (see Figure 1). To illustrate our model, consider the sample network depicted in Figure 1, in which each peer has a several neighbors and ultra-peers. Peer 11 has Peer 2, Peer 5, Peer 15 and Peer 17 as neighbors; and Peer 11 as ultra-peer.

In particular, our distributed discovery mechanism exploits metrics to broadcast queries in the P2P network, by associating a TTL[1] and $\Psi(\alpha, \lambda)$[2] value to the query messages; bigger values increase the success rate but may quickly lead to P2P network congestion. We then use this analysis to propose an algorithm that seeks to hit the minimum number of peers necessary to obtain the desired number of results for a given search. The general algorithm is described as follows.

Algorithm 1 : Distributed Discovery of SWs ();

1: **Input:** Upon reception of user request (R) at peer such as Search the SWs:(Input, Output, Precondition, Result and TextDescription).
2: **Output:** A set of SWs which responds to the user request.
3: **Begin**
4: **Discovery-Matching ();**
5: **if (There is a local SWs) then** // to discover a local SWs in this peer
6:     Send the Favorable Response;
7: **Else**
8: Calculate the TTL;
9: Calculate the $\Psi(\alpha, \lambda)$;
10: **if** (TTL $\geq$ 1) And ($\Psi(\alpha, \lambda) \geq Threshold$) **then**
11:     $TTL \leftarrow TTL - 1$;
12: **P2P-Discovery ();** // to propagate the query in ultra-peers or neighbors
13: **Else**
14: Drop the request;
15: **EndIf**
16: **EndIf**
17: **End**

Each peer in the unstructured P2P network executes the following main algorithm when receiving a request. Initially, each peer executes the main algorithm from step 1 to step 4 where it tries to response to the request locally (local basic SWs) and it sends the response to the transmitting peer (step 5 and step 6 of the main algorithm). If there is not a possibility to answer the request locally, the peer starts a distributed P2P discovery (step 12 of the main algorithm).

In addition, if the peer has no positive results, it decreases the TTL value by 1 and confirms $\Psi(\alpha, \lambda) \geq Threshold$, then it sends query towards all its immediate ultra-peers with the TTL value. The ultra-peer then executes the operations of TTL and $\Psi(\alpha, \lambda)$, to propagate the query to its other ultra-peers. If the ultra-peer has no positive results, then the ultra-peer forward the message contains the new TTL value to its ultra-peers. If all ultra-peers fail, the request will be forwarded to the next neighbors where the process is repeated until all neighbors are accessed or a positive result is given. If no positive result is found, the result sent by the last peer consultation will be negative.

---

[1] **TTL :** Time to Live.
[2] Gamma function.

One of the major issues in our approach is the ability to determine the degree of similarity between the concepts of two OWL-S process models. We investigate the use of hybrid matching component to increase the number of matching between the requests and OWL-S process model available in the OWL-S repository of different peers in the P2P network to improve system responses by the system that is asked. Each peer implements a number of SWs described semantically with OWL-S process model for publishing and discovering SWs in a local repository. The current trend of most matching systems is to combine the different matching techniques of ontologies OWL-S. The rationale behind the application of hybrid matching techniques in the scenario considered here is that we need two kinds of matching information to label semantic similarity links, in order to support efficient and effective SWs discovery.

We say that there is a similarity between concepts of Request R and Web service S if there is a degree of match between the set of concepts annotating the parameter (*Input*, *Output*, *Precondition*, *Result* and *TextDescription*) of Request R and the set of concepts annotating the same parameter (*Input*, *Output*, *Precondition*, *Result* and *TextDescription*) of Web service S.

To do so, we develop the procedure **Discovery-Matching()** (See algorithm 1) which creates a vector of semantic links in order to solve a service discovery request. The vector of semantic links stores the semantic similarity links established between the different SWs in a peer of the P2P network and the user request. In the remaining, we use the well-known similarity metric to compute the similarity assessment [10]:

$$Sim(R, S) = 1 - [2 \times \omega(lca(R, S)) - \omega(R) - \omega(S)] \quad (1)$$

where $lca(R, S)$ is the least common ancestor of R and S, and $\omega(R/S) \in [0, 1]$ is the weight of the concept R (or S) in the ontology. The similarity between two sets of concepts is then the average similarity of the mapping that maximizes the sum of similarities between the concepts of these sets. However, the similarity between $R = (In_R, Out_R, Pre_R, Res_R, Tex_R)$ and $S = (In_S, Out_S, Pre_S, Res_S, Tex_S)$ is the weighted average of their *Input*, *Output*, *Precondition*, *Result* and *TextDescription* similarities, as formalized by evaluating the following function [3]:

$$Similarity(R, S) = (\omega_{In} \times Sim(In_R, In_S) + \omega_{Out} \times Sim(Out_R, Out_S) + \omega_{Pre} \times Sim(Pre_R, Pre_S) + \omega_{Res} \times Sim(Res_R, Res_S) + \omega_{Tex} \times Sim(Tex_R, Tex_S)) \quad (2)$$

Where $Sim(R, S)$ is the function computing the similarity between two sets of parameters. Weights ($\omega_{In}, \omega_{Out}, \omega_{Pre}, \omega_{Res}$ and ω_{Tex}) designate, respectively the weight of *Input*, *Output*, *Precondition*, *Result* and *TextDescription* to evaluate the similarity ($\omega_{In} + \omega_{Out} + \omega_{Pre} + \omega_{Res} + \omega_{Tex} = 1$). This measure of similarity is probably the most widely used today and the most effective way to determine the semantic proximity between two concepts. This measure is used due to its high performances for process model matchmaking.

In our P2P approach, to find a resource, a peer broadcast a message to its ultra-peers or neighbors. However, If all remote peers sends the query by flooding, than a serious problem of flooding emerges due to the excessive traffic overheads caused by a large number of redundant message forwarding, increasing the bandwidth consumption and reducing the performance of the P2P system; particularly in a P2P system with a high connectivity topology. Although it is effective in our P2P approach, flooding is very inefficient because it results in a great amount of redundant messages. To optimize the flooding, some criteria have to be defined to select the peer for which the request will be sent. Each peer must limit the number of simultaneous transfers of request.

Typically two connected peers each ensures the other to be active in the P2P network and begins sending requests with the maximum interval of time defined by the session time. The session time of peer could be modeled by an exponential distribution. In this work, our aim is to achieve an extremely high network scalability and enhancing the other network performance metrics. For this purpose, we analyse how this widespread property of P2P networks can be more accurately described by Gamma Distribution that dynamically adapts the network effective density to enable a large scale deployment of the P2P network while providing good overall performances.

From the function computing the similarity between two sets of concepts $Similarity(R, S) \in [0, 1]$ and given peer TTL value, we have defined the following evaluating Gamma Distribution function as follows:

$$P(session < t) = \Psi(t|\alpha, \lambda) = \frac{\lambda^{\alpha}}{\Gamma(\alpha)} \times \exp^{-(\frac{t}{\lambda})} \tag{3}$$

In a P2P network of N peers with D being the average peer degree, we experiment with several values of the TTL to describe the shape α and the scale λ parameters when:

$$\Gamma(\alpha) = \int_{o}^{\infty} x^{(\alpha-1)} \exp^{-x} dx; \alpha = \frac{\ln(N-1)}{D^{TTL}}; \lambda = \frac{1}{TTL+1} \tag{4}$$

We opt for a Gamma function that uses exponential function ($\exp^{-(\frac{t}{\lambda})}$) because a high Gamma function incurs more redundant rebroadcast while a low Gamma function leads to low reachability. Moreover, peers with low values of TTL, α and λ should be assigned a high Gamma function while those with high values of TTL, α and λ are assigned a low Gamma function. Therefore, as the number of neighbors increases, the Gamma function should decreases.

Our motivation for this Gamma function is to enhance rebroadcast decision by taking into account key network parameters and peer information through TTL, α and λ value. We implement our matching technique for Gamma function that can be used to assess the similarity function ($Similarity(R, S) \in [0, 1]$) between two concepts of OWL-S process models and to provide an alignment between them; 0 means the concepts are totally different, 1 means that they are totally similar. Consequently, the similarity between concepts of Request R and Web service S is defined on the basis of their semantic relationship in the ontology.

We apply results achieved by statistical analysis of random graphs to selecting appropriate initial value for TTL of the query. Applying the results provided by [9], each peer decides on the appropriate TTL value for its queries based on the information collected locally. The TTL between two randomly chosen peers on any P2P network is approximated as follows:

$$\mathbf{TTL} = \frac{\ln[(N-1)(Z_2 - Z_1) + Z_1^2] - \ln(Z_1)^2}{\ln(Z_2/Z_1)} \tag{5}$$

Where Z_i is the number of neighbors which are i hops away from the originator peer. TTL between two peers presents a reasonable estimation of the distance between the originator of the query and the peer that eventually serves the requested object. TTL is actually the scope of the request: more important it is, the more there will be peers that will be visited, and the request will likely be satisfied. A large TTL also causes average response greater. Using a large TTL, this type of infrastructure can meet a maximum of elements corresponding to the search criteria. In addition, this approach is fast and reliable (it behaves very well in highly dynamic networks with many arrivals or departures of peers). In the next section we will proved our approach, evaluation results and share our experiences.

4 Experiments and Results

We conducted a series of experiments to demonstrate and evaluate the effectiveness and the efficiency of our scalable approach for discovering SWs in the unstructured P2P network through event-driven simulations, which are usually used to evaluate the performance of large-scale P2P systems. For this reason, we are using PeerSim simulator [8] to simulate an unstructured P2P network. The experimentation has been performed with the number of peers that varies in the range from 100 to 1500 with an iteration range of 200. By generated requests we mean the total number of overall requests produced and forwarded on the unstructured P2P network as a consequence of a request submitted to a peer; this parameter depends on the number of peer in the network and on the peers average number of connections to its neighbors. We run our experiments on computers with Intel Core i5 CPU (2.54 GHz and 4GB RAM) under Windows 7. The data used in our experiments are OWLS-TC[3] 4.0. It provides 1083 SWs written in OWL-S 1.1. It provides a set of 42 test queries which are associated with relevance sets to conduct performance evaluation experiments. With regard to the different similarity measures which are implemented in our approach, we used the Java API SIMPAC[4] (Similarity Package) which represents a comprehensive library that contains all the important similarity measures. We use JWordNetSim to measure the similarity between synsets in WordNet[5] 2.0.

[3] http://www.semwebcentral.org/projects/owls-tc/
[4] http://sourceforge.net/projects/simmetrics
[5] http://wordnet.princeton.edu/

To evaluate the efficiency and scalability of our approach, we must compare our approach with a simple flooding protocol Gnutella P2P protocol [1]. In our experiment, we study the significance of our algorithm in terms of computation time. We define the optimality ratio as follows :

$$\textbf{Optimality Ratio} = \frac{W_{Gamma} - W}{W_{Gamma}} \tag{6}$$

where W is the execution time of our algorithm without Gamma function and W_{Gamma} is the execution time of our algorithm by applying Gamma function. The execution time has been measured upon the number of invokes that our approach, reflecting the number of discovering SWs in unstructured P2P network. This helps demonstrate the scalability of our approach. The execution time has been measured using The Eclipse Test and Performance Tools Platform[6]. Experimentation results will be analyzed in the following.

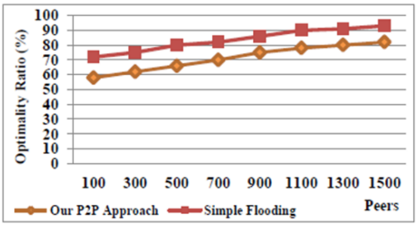

Fig. 2. Optimality Ratio vs the Peer.

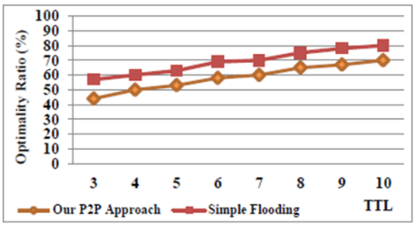

Fig. 3. Optimality Ratio vs the TTL.

The graph in Figure 2 or 3 illustrates some of the performance test results. We notice that the optimality ratio increases slightly along with the number of peers or TTL. The results reported in these figures represent the average computation times found for each given request. Despite the exponential theoretical complexity, the figure 2 or 3 shows that our scalable algorithm can be used, with acceptable computation time. More clearly shown in Figure 2 or 3, Gnutella protocol has a very high computation time because it uses simple flooding algorithm. The simple flooding algorithm leads to high computational time. Our P2P approach performs better than the Gnutella protocol in terms of optimality ratio because it uses Gamma function that reduces considerably the computation time.

---

[6] TPTP: http://www.eclipse.org/tptp/

5 Conclusion and Future Work

In this paper, we have discussed efficient and effective approach, for improving the performance of distributed and cooperative discovery of SWs in the unstructured P2P networks. We have proposed the matching technique of OWL-S to fulfill the users requirements and the Gamma Distribution which reduces the query traffic. In the future work, we will focus our efforts on optimizing the communicational complexity by using optimization techniques such as heuristics and meta heuristics.

References

1. Hassan Barjini, Mohamed Othman, Hamidah Ibrahim, and Nur Izura Udzir. Hybridflood: minimizing the effects of redundant messages and maximizing search efficiency of unstructured peer-to-peer networks. *Cluster Computing*, 17(2):551–568, 2014.
2. Devis Bianchini, Valeria De Antonellis, and Michele Melchiori. P2p-sdsd: on-the-fly service-based collaboration in distributed systems. *International Journal of Metadata, Semantics and Ontologies*, 5(3):222–237, 2010.
3. Adel Boukhadra, Karima Benatchba, and Amar Balla. Hybrid ontology-based matching for distributed discovery of sws in p2p systems. In *16th IEEE International Conference on High Performance Computing and Communication, HPCC 2014, Paris, France, August 20-22, 2014*, pages 116–123. IEEE, 2014.
4. Adel Boukhadra, Karima Benatchba, and Amar Balla. Efficient P2P approach for a better automation of the distributed discovery of sws. In *Distributed Computing and Artificial Intelligence, 12th International Conference, DCAI 2015, Salamanca, Spain, June 3-5, 2015*, pages 277–284, 2015.
5. Adel Boukhadra, Karima Benatchba, and Amar Balla. Efficient distributed discovery and composition of OWL-S process model in P2P systems. *J. Ambient Intelligence and Humanized Computing*, 7(2):187–203, 2016.
6. P Fragopoulou, C Mastroianni, R Montero, A Andrjezak, and D Kondo. Self-* and adaptive mechanisms for large scale distributed systems. In *Grids, P2P and Services Computing*, pages 147–156. Springer, 2010.
7. Angelo Furno and Eugenio Zimeo. Efficient cooperative discovery of service compositions in unstructured p2p networks. In *Parallel, Distributed and Network-Based Processing (PDP), 2013 21st Euromicro International Conference on*, pages 58–67. IEEE, 2013.
8. Alberto Montresor and Márk Jelasity. Peersim: A scalable p2p simulator. In *Peer-to-Peer Computing, 2009. P2P'09. IEEE Ninth International Conference on*, pages 99–100. IEEE, 2009.
9. Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical review E*, 64(2):026118, 2001.
10. Pavel Shvaiko and Jérôme Euzenat. Ontology matching: state of the art and future challenges. *Knowledge and Data Engineering, IEEE Transactions on*, 25(1):158–176, 2013.
11. Huayou Si, Zhong Chen, Yong Deng, and Lian Yu. Semantic web services publication and oct-based discovery in structured p2p network. *Service Oriented Computing and Applications*, 7(3):169–180, 2013.

http://www.springer.com/978-3-319-48828-8

Intelligent Distributed Computing X
Proceedings of the 10th International Symposium on Intelligent Distributed Computing – IDC 2016, Paris, France, October 10-12 2016
Badica, C.; El Fallah Seghrouchni, A.; Beynier, A.; Camacho, D.; Herpson, C.; Hindriks, K.; Novais, P. (Eds.)
2017, XIII, 248 p. 68 illus., Hardcover
ISBN: 978-3-319-48828-8