

# Stigmergic Modeling for Web Service Composition and Adaptation

Ahmed Moustafa<sup>1</sup>, Minjie Zhang<sup>1</sup> and Quan Bai<sup>2</sup>

<sup>1</sup>School of Computer Science and Software Engineering  
University of Wollongong, Wollongong, NSW 2522, Australia  
{aase995,minjie}@uow.edu.au

<sup>2</sup>School of Computing and Mathematical Sciences  
Auckland University of Technology, Auckland, New Zealand  
quan.bai@aut.ac.nz

**Abstract.** As Web services become widespread, many complex applications require service composition to cope with high scalability and heterogeneity. Centralized Web service composition approaches are not sufficient as they always limit the scalability and stability of the systems. How to efficiently compose and adapt Web services under decentralized environments has become a critical issue, and important research question in Web service composition. In this paper, a stigmergic-based approach is proposed to model dynamic interactions among Web services, and handle some issues in service composition and adaptation. In the proposed approach, Web services and resources are considered as multiple agents. Stigmergic-based self-organization among agents are adopted to evolve and adapt Web service compositions. Experimental results indicate that by using this approach, service composition can be efficiently achieved, despite dealing with incomplete information and dynamic factors in decentralized environments.

*Keywords:* Web Services, Decentralized Composition, Dynamic Adaptation.

## 1 Introduction

In recent years, service composition and adaptation has drawn the interests of researchers in a number of fields. Generally, service composition is to fulfil users' requirements by combining services in different repositories. Those requirements can be functional or non-functional. The functional requirements are in relation to targeting the overall outcome concerning the the underway business process, while the non-functional requirements pertain to the quality of the composition as a whole in terms of availability, reliability, cost and response time. Service adaptation is to adjust service "behaviors" and/or composition strategies to adapt to the changes of users' requirements, *QoS* fluctuations and other prospective runtime dynamics.

Although in the Web service domain, many standards have been developed and adopted for facilitating service composition, lack of supports for decentralized working environments is still a standing challenge and an obvious drawback of many existing composition approaches. Hence, efficient Web service applications need not only to be managed from a centralized perspective, but also to incorporate decentralized scenarios.

Stigmergic interaction, exhibited by social insects to coordinate their activities, has recently inspired a vast number of computer science applications [7, 9]. In these works, intelligent components (i.e., agents) interact by leaving and sensing artificial pheromones (i.e., markers) in a virtual environment. Such pheromones can encode and describe application-specific information to be used to achieve specific tasks. From a general perspective, stigmergic interactions have two major advantages: (i) Stigmergic interactions are mediated by pheromones and are completely decoupled. This feature is suitable for open and dynamic environments. (ii) Stigmergic interaction naturally supports application-specific context awareness, in that pheromones provide agents with an application-specific representation of their operational environment (e.g., in ant foraging, pheromones provide a representation of the environment in terms of paths leading to food sources).

In this paper, we propose a stigmergic-based approach for modeling Web service compositions and adaptations. The proposed approach exhibits the concept of decentralization, and allows Web services to compose and adapt in open decentralized environments. The rest of this paper is organized as follows. The principle and introduction of the stigmergic-based service composition and adaptation approach is introduced in Section 2. In Section 3, some experimental results are presented for evaluating the proposed approach. Section 4 gives a brief review of related work and discussions. Finally, the paper is concluded in Section 5.

## 2 Stigmergic Service Composition and Adaptation

In this section, an approach named Stigmergic Service Composition and Adaptation is proposed to address the problem of Web service composition and adaptation in decentralized environments. The proposed approach is inspired by the concept of *Digital Pheromone* [12], which is a stigmergic-based mechanism for coordinating and controlling swarming objects. Examples from natural systems show that stigmergic systems can generate robust, complex, intelligent behaviors at the system level even when the individual agents only possess very limited or even no intelligence. In these systems, intelligence resides not in a single distinguished agent (as in centralized control) nor in each individual agent, but in the whole group of agents distributed in the environment [3, 10].

### 2.1 Web Service Composition/Adaptation Modeling

In our approach, each Web service is envisaged as a service agent, and a Web service system can then be considered as a multi-agent system with a num-

ber of interactive service agents. These agents achieve collaborations and self-organizations via exchanging “pheromone” and performing several pheromone operations. A service is composed if a group of service agents can form an organization to collaborate.

**Definition 1: Service Agent.** A *Service Agent*  $p_i$  is defined as a tuple  $p_i = \langle id, F \rangle$ , where  $id$  is the identifier of the service agent,  $F$  is the pheromone store for facilitating composition and decentralized self-coordination (as detailed in Definition 2)

**Definition 2: Pheromone Store.** A *Pheromone Store*  $F$  is a set of pheromone, i.e.,  $F = \{f_1, f_2, \dots, f_n\}$ . Each pheromone flavor  $f_j$  holds a scalar value which represents the trail of a certain service agent.

Based upon user requirements, a service agent  $p_i$  requests the composition of an abstract workflow  $r_j$  such that  $r_j = \{r_{j1}, r_{j2}, \dots, r_{jn}\}$ , where  $r_{ji}$  represents a concrete service or a resource needs to be composed for satisfying a single unit of functionality or a subtask. To this end, the requester service agent traverses a network of directly linked service agents, and look for suitable services and resources for constructing workflow  $r_j$ . Once constructed, the requester service agent deposits/withdraws certain amount (i.e.,  $Q_{f_j}$ ) of pheromone flavor  $f_j$  into/from the digital pheromone stores of all provider service agents, which took part in this composition round. The decision of depositing or withdrawing depends on the quality of the provided service. Based upon the strength of the trail of this pheromone flavor  $f_j$ , in future rounds, similar service requests will be guided towards the highest quality workflow quickly without the need to traverse the entire network again. Namely, even in open decentralized environments, a requester service agent can immediately make decisions based on the pheromone trail left by other requester service agents.

## 2.2 Pheromone Operations

In this approach, we define three primary pheromone operations for coordinating service agents, and enabling them to achieve self-organizations.

**Pheromone Aggregation:** A *Digital Pheromone*  $Q_{f_j}$  can be deposited and withdrawn from any service agent  $p_i$ . Deposits of a certain flavor  $f_j$  are added to the current amount of that flavor of pheromone  $s(Q_{f_j}, p_i, t)$  located at that service agent  $p_i$  and time  $t$ .

**Pheromone Evaporation:** A *Pheromone Evaporation* is the operation wherein a pheromone flavor  $f_j$  evaporates over time  $t$ . This operation will generate an Evaporation Factor  $E_{f_j}$  ( $0 < E_{f_j} \leq 1$ ) to weaken obsolete information.  $E_{f_j}$  can be obtained via Equation 1:

$$E_{f_j} = e^{\frac{-\Delta t(i)}{\lambda}} \quad (1)$$

where  $trianglet(i)$  is the time difference between the current time and when  $f_i$  has been left, and  $\lambda$  is the parameter to control the evaporation speed.

**Pheromone Propagation:** A *Pheromone Propagation*  $G_{f_j}$  is the operation wherein a pheromone flavor  $f_j$  propagates from a service agent  $p_i$  to service agent  $p_k$  based upon a neighborhood relation  $N$  where  $p_k \in N(p_i)$ . The act of propagation causes pheromone gradients  $g(Q_{f_j}, p_i, t)$  to be formed.

### 2.3 Dynamic Coordination and Adaptation

The three operations introduced in the above subsection serve as a basis for the coordination mechanism. By using these operations, pheromone can be exchanged and circulated among distributed service agents. Furthermore, a service composition and adaptation algorithm (i.e., Algorithm 1) is designed to enable agents to learn the best composition path, and to adapt to the changing environment. The underlying proposed algorithm rests on two fundamental equations to enhance the self-organization behavior, i.e., Equation 2 and Equation 3.

Let  $t$  represents an update cycle time, which is the time interval between propagation, pump auto-deposits, and evaporation of a certain flavor  $f_j$ ;  $P = \{p_1, p_2, \dots, p_n\}$  represent a set of service agents;  $N : P \rightarrow P$  represents a neighbor relation between service agents;  $s(Q_{f_j}, p_i, t)$  represents the strength of pheromone flavor  $f_j$  at service agent  $p_i$  and time  $t$ ;  $d(Q_{f_j}, p_i, t)$  represents the sum of external deposits of pheromone flavor  $f_j$  within the interval  $(t-1, t]$  at service agent  $p_i$ ;  $g(Q_{f_j}, p_i, t)$  represents the propagated input of pheromone flavor  $f_j$  at time  $t$  to service agent  $p_i$ ;  $E_{f_j} \in (0, 1)$  represents the evaporation factor for flavor  $f_j$ ;  $G_{f_j} \in (0, 1)$  represents the propagation factor for flavor  $f_j$ ;  $T_{f_j}$  represents the threshold below which  $S(Q_{f_j}, p_i, t)$  is set to zero.

The evolution of the strength of a single pheromone flavor at a given service agent is defined in Equation 2:

$$s(Q_{f_j}, p_i, t) = E_{f_j} * [(1 - G_{f_j}) * (s(Q_{f_j}, p_i, t-1) + d(Q_{f_j}, p_i, t)) + g(Q_{f_j}, p_i, t)] \quad (2)$$

where  $E_{f_j}$  is the evaporation factor pheromone flavor  $f_j$  (refer to Equation 1),  $1 - G_{f_j}$  calculates the amount remaining after propagation to neighboring service agents,  $s(Q_{f_j}, p_i, t-1)$  represents the amount of pheromone flavor  $f_j$  from the previous cycle,  $d(Q_{f_j}, p_i, t)$  represents the total deposits made since the last update cycle (including pump auto-deposits) and  $g(Q_{f_j}, p_i, t)$  represents the total pheromone flavor  $f_j$  propagated in from all the neighbors of  $p_i$ . Every service agent  $p_i$  applies this equation to every pheromone flavor  $f_j$  once during every update cycle.

The propagation received from the neighboring service agents is described in Equation 3:

$$g(Q_{f_j}, p_i, t) = \sum_{p_k \in N(p_i)} \frac{G_{f_j}}{N(p_k)} (s(Q_{f_j}, p_k, t-1) + d(Q_{f_j}, p_k, t)) \quad (3)$$

In Equation 3, it can be found that each neighbor service agent  $p_k$  ( $p_k \in N(p_i)$ ) propagates a portion of its pheromone to  $p_i$  in each update cycle  $t$ . This

portion depends on the parameter  $G_{f_j}$  and the total number of  $p_k$  neighbors  $N(p_k)$ .

Using Equations 2 and 3, we can demonstrate several critical stability and convergence rules, including Local Stability, Propagated Stability and Global Stability.

**Rule 1: Local Stability.** A *Local Stability*  $S_l$  is the rule in which the strength of the output propagated from any set of service agents  $M \subset P$  to their neighbors  $N \subset P$  at  $t + 1$  is strictly less than the strength of the aggregate input (external plus propagated) to those service agents at  $t$ .

**Rule 2: Propagated Stability.** A *Propagated Stability*  $S_p$  is the rule in which there exists a fixed upper limit  $\alpha$  to the aggregated sum of all propagated inputs at an arbitrary service agent  $p_i$  if one-update cycle  $t$  and one-service agent  $p_i$  external input is assumed.

**Rule 3: Global Stability.** A *Global Stability*  $S_g$  is the rule in which the pheromone strength  $s(Q_{f_j}, p_i, t)$  in any place agent is bounded. If the pheromone strength drops below threshold  $T_{f_j}$ , it disappears from the pheromone store,  $s(Q_{f_j}, p_i, t) \geq T_{f_j}$ .

The Stigmergic Service Composition and Adaptation can be represented by Algorithm 1.

---

**Algorithm 1** Stigmergic Service Composition and Adaptation Algorithm

---

```

for each  $p_i \in P$  do
  while  $s(Q_{f_j}, p_i, t) \geq T_{f_j}$  do
    for  $t = 1 : n$  do
       $g(Q_{f_j}, p_i, t) = \sum_{p_k \in N(p_i)} \frac{G_{f_j}}{N(p_k)} (s(Q_{f_j}, p_k, t - 1) + d(Q_{f_j}, p_k, t));$ 
      if  $g(Q_{f_j}, p_i, t) < \alpha$  then
         $g(Q_{f_j}, p_i, t) = g(Q_{f_j}, p_i, t);$ 
      else
         $g(Q_{f_j}, p_i, t) = \alpha;$ 
      end if
       $p_i^{in} = g(Q_{f_j}, p_i, t);$ 
       $p_i^{out} = (G_{f_j}) * (s(Q_{f_j}, p_i, t - 1) + d(Q_{f_j}, p_i, t));$ 
      while  $p_i^{out} < p_i^{in}$  do
         $s(Q_{f_j}, p_i, t) = E_{f_j} * [(1 - G_{f_j}) * (s(Q_{f_j}, p_i, t - 1) + d(Q_{f_j}, p_i, t)) + g(Q_{f_j}, p_i, t)];$ 
      end while
    end for
  end while
end for

```

---

Using Algorithm 1, the Stigmergic Service Composition and Adaptation approach can account for the decline and emergence of a set of service agents having the same pheromone flavor. By ever choosing the trail with the strongest flavor, the proposed approach can adapt to the dynamics of open runtime environments

by employing new emerging service agents with stronger flavors and forgetting other service agents with fading flavors.

### 3 Experimentation and Results

#### 3.1 Experiment Setup

Three experiments have been conducted to assess the proposed Stigmergic Service Composition and Adaptation approach. The proposed approach runs in successive iterations till reaching a convergence point. The proposed approach converges to a near optimal solution once it receives the same approximate value of accumulative  $QoS$  for a number of successive iterations. Those accumulated  $QoS$  values are compared iteration by iteration and the difference is projected against a threshold. For our experiments, this threshold value is set to 0.0001, and the number of successive iterations is set to 100. The first experiment examines the effectiveness of the proposed approach in composing Web services under a decentralized environment. The second experiment explores the speed of composing Web services using the proposed approach under a decentralized environment, whereas the third experiment inspects the performance and the efficiency of the proposed approach related to the scale of the environment. All the three experiments have been conducted on 3.33 GHz Intel core 2 Duo PC with 3 GB of RAM.

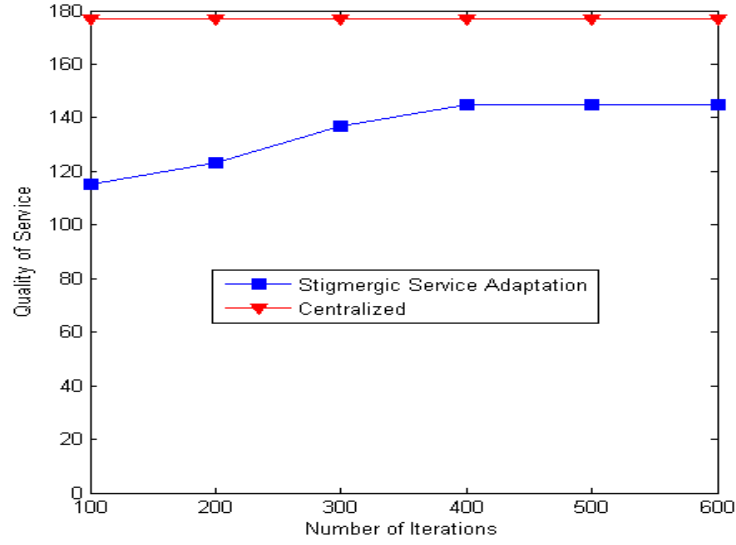
#### 3.2 Result Analysis

The results of the three experiments are demonstrated and analyzed in details in the following three subsections, respectively.

**Experiment 1: Decentralized Environment** To explore the effectiveness of the proposed Stigmergic Service Composition and Adaptation approach under decentralized environments, a composition request has been submitted to the proposed approach and the  $QoS$  values of the composed workflow is contrasted to those values obtained by a centralized service composition approach. The proposed approach assumes no prior knowledge of the environment including available service agents and their  $QoS$  values, while other centralized approaches assume the existence of a central service registry in which all the the functional and  $QoS$  qualities of Web services are recorded.

Table 1 shows the progression of the  $QoS$  values of the composed workflow using the proposed approach against those values obtained by centrally composing the same workflow using the centralized approaches, e.g., Critical Path Method (CPM). Those values are recorded on a 100-iteration basis.

Figure 1 depicts the contrast between the  $QoS$  values obtained form the proposed approach against those values obtained from a centralized approach ( $y$  axis) and the corresponding number of iterations ( $x$  axis). As shown in Figure 1, the centralized service composition approach shows a steady line with a  $QoS$



**Fig. 1.** Decentralized Environment

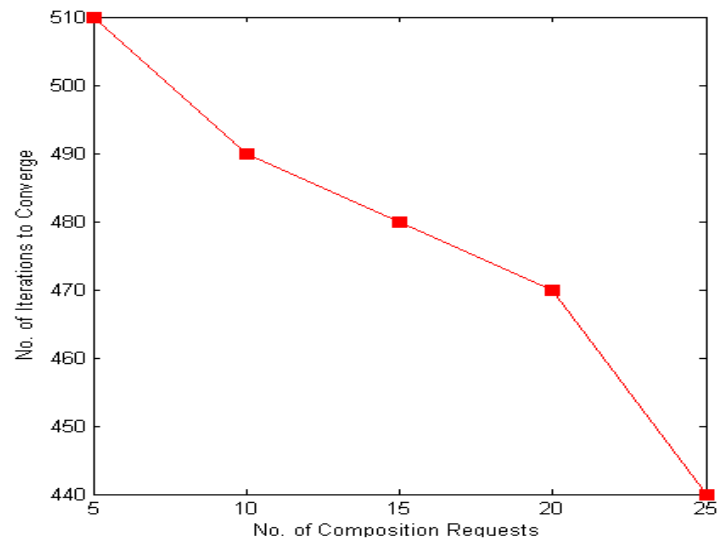
**Table 1.** Decentralized Environment

Iterations	Stigmergic Service Composition and Adaptation	Centralized(CPM)
100	115	177
200	123	177
300	137	177
400	145	177
500	145	177
600	145	177

value of 177 regardless of the number of iterations. This is attributed to the assumption of a prior knowledge of the available Web services and their qualities, which is not the case in real world. On the other hand, the proposed Stigmergic Service Composition and Adaptation approach shows a progressive line representing an upward trend in the  $QoS$  values obtained as the number of iterations increases. Due to its decentralized nature, the proposed approach needs to run for a number of iterations before showing stability in results, which happens in our experiment when the number of iterations goes beyond 300. The best stable  $QoS$  obtained is 145, which shows 82% rate of effectiveness compared to the optimal value of 177 in a fully centralized approach. This result demonstrates a good performance considering a decentralized open environment with no prior knowledge of available Web services and their  $QoS$  values.

**Experiment 2: Composition Speed** To examine the composition speed of the Stigmergic Service Composition and Adaptation approach, a varying number of composition requests having the same composition goal have been submitted against an environment of 10000 service agents. Those composition requests supposedly have different starting points and each time the number of iterations to converge to the composition goal is measured.

Table 2 shows a range of composition requests and the number of iterations to converge.



**Fig. 2.** Composition Speed

**Table 2.** Composition Speed

No. of Composition Requests	No. of Iterations to Converge
5	510
10	490
15	480
20	470
25	440

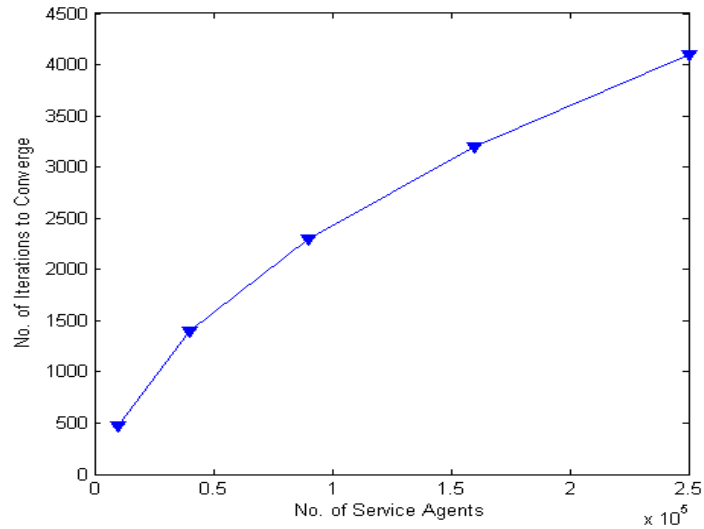
Figure 2 depicts the relationship between the number of composition requests ( $x$  axis) and the corresponding number of iterations to converge ( $y$  axis). In Fig-



ure 2, the trend line shows a downward movement describing a linear relationship. As the number of running composition requests increases, the number of iterations to converge to the composition goal decreases. This explains the benefit of the accumulated pheromone trails left by previous service agents working for the same composition goal. The service agents in new iterations have higher probabilities to make benefit of the pheromone flavor trails left by previous service agents used by other composition requests for the same composition goal, thus to follow their trail to fulfill that goal faster. As the number of running composition requests increases, the pheromone trails left by service agents employed by those composition requests become more strengthened and thus the potentiality to guide new service agents working towards the same composition goal increases.

**Experiment 3: Environment Scale** To examine the efficiency of the proposed approach under large scale open environments, twenty composition requests sharing the same composition goal have been submitted against a range of environment scales, expressed in terms of varying numbers of service agents. The composition requests supposedly have different starting points and each time the number of iterations to converge to the composition goal is measured.

Table 3 shows a range of environment scales with the number of iterations to converge.



**Fig. 3.** Environment Scale

**Table 3.** Environment Scale

No. of Service Agents	No. of Iterations to Converge
10000	470
40000	1400
90000	2300
160000	3200
250000	4100

Figure 3 depicts the relationship between the scale of the environment in terms of the number of service agents ( $x$  axis) and the corresponding number of iterations needed to converge ( $y$  axis). The trend line shows an upward movement describing a linear relationship. This linear relationship proves the efficiency of the proposed approach in composing and adapting Web services in open large scale environments, e.g., the Internet. As the scale of the environment increases, the number of iterations to converge increases linearly not exponentially accordingly. This shows the robustness of the proposed approach under large scale service environments, e.g., the Internet.

## 4 Related Work and Discussions

Various approaches have been proposed to handle the problem of dynamic composition of Web services [8, 13, 4]. However, most of these approaches are based on a centralized architecture. In these models, a service requester submits requests to a centralized directory, and the directory decides which service on the list should be returned to the requester. This assumption on centralized control becomes impractical in scenarios with dynamic arrivals and departures of service providers, which require frequent updates of the central entities, resulting in a large system overhead. In contrast, our proposed approach can work under a fully decentralized environment where no central directory exists. The service agent has to find other matching service agents during run time based on neighborhood relationships and track for their  $QoS$  by exchanging pheromone flavors. This enables our approach to adapt effectively under dynamic environments accounting for the arrival and departure of service providers at any time.

Some approaches exploiting distributed techniques have already been proposed in the literature [6, 11, 5]. As the first step towards decentralization of service composition, these approaches introduced a distributed directory of services. For example, Kalasapur *et al.* [5] proposed a hierarchical directory in which the resource-poor devices depend on the resource-rich devices to support service discovery and composition. However, this approach is not fully decentralized. There exist nodes that perform the task of service discovery and composition for other nodes. In contrast, our approach is fully decentralized as a set of service agents cooperate together coherently to fulfill the composition request.

Another important approach was proposed by Basu *et al.* [1] in which a composite service is represented as a task-graph and subtrees of the graph are com-

puted in a distributed manner. However, this approach assumes that a service requestor is one of the services and the approach relies on this node to coordinate service composition. A similar approach was proposed by Drogoul [2], whose work is different from Basu *et al.* [1] in terms of the way of electing the coordinator. For each composite request, a coordinator is selected from within a set of nodes. The service requestor delegates the responsibility of composition to the elected coordinator. Both approaches [2, 1] exhibit some limitations. Although service discovery is performed in a distributed manner, service composition still relies on a coordinator assigned for performing the task of combining and invoking services. Also, both of the two approaches assume the direct interaction between nodes responsible for service discovery and service composition. In contrast, our approach overcomes those limitations as follows. (1) In our approach, there is no special entity to manage service composition process; (2) Service providers communicate only with their local neighbors where no service provider knows the full global information; and (3) Our approach adopts an indirect interaction scheme where the set of service agents responsible for service composition and adaptation can interact with each other through the environment by exchanging pheromone trails.

## 5 Conclusion

This paper focuses on modeling Web service composition and adaptation under decentralized dynamic environments. The paper presented a service composition and adaptation approach using stigmergic interactions to actively adapt to dynamic changes in complex decentralized Web service composition environments. The proposed approach is able to adapt actively to QoS fluctuations considering both potential degradation and emergence of *QoS* values. The experimental results have shown the effectiveness of the proposed approach in highly complex and dynamic Web service composition environments. The future work is set to investigate the performance of the proposed approach in real world applications and to study the potentiality of building a web service trust model using the proposed approach.

## References

1. P. Basu, Wang Ke, and T.D.C. Little. A novel approach for execution of distributed tasks on mobile ad hoc networks. In *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, volume 2, pages 579 – 585 vol.2, mar 2002.
2. Dipanjan Chakraborty, Anupam Joshi, Tim Finin, and Yelena Yesha. Service composition for mobile environments. *Mob. Netw. Appl.*, 10(4):435–451, August 2005.
3. J.N. Haack, G.A. Fink, W.M. Maiden, A.D. McKinnon, S.J. Templeton, and E.W. Fulp. Ant-based cyber security. In *Information Technology: New Generations (ITNG), 2011 Eighth International Conference on*, pages 918 –926, april 2011.

4. O. Hatzi, D. Vrakas, M. Nikolaidou, N. Bassiliades, D. Anagnostopoulos, and I. Vlahavas. An integrated approach to automated semantic web service composition through planning. *Services Computing, IEEE Transactions on*, PP(99):1, 2011.
5. S. Kalasapur, M. Kumar, and B.A. Shirazi. Dynamic service composition in pervasive computing. *Parallel and Distributed Systems, IEEE Transactions on*, 18(7):907–918, july 2007.
6. Chun-Feng Liao, Ya-Wen Jong, and Li-Chen Fu. Toward reliable service management in message-oriented pervasive systems. *Services Computing, IEEE Transactions on*, 4(3):183–195, july-sept. 2011.
7. Chuanjun Ren, Hongbing Huang, and Shiyao Jin. Stigmergic collaboration: A sign-based perspective of stigmergy. In *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, volume 3, pages 390–393, may 2010.
8. Xiaoqiang Ren and Ruilian Hou. Extend uddi using ontology for automated service composition. In *Mechanic Automation and Control Engineering (MACE), 2011 Second International Conference on*, pages 298–301, july 2011.
9. D. Rosen and D.D. Suthers. Stigmergy and collaboration: Tracing the contingencies of mediated interaction. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pages 1–10, jan. 2011.
10. O. Simonin, T. Huriaux, and F. Charpillet. Interactive surface for bio-inspired robotics, re-examining foraging models. In *Tools with Artificial Intelligence (IC-TAI), 2011 23rd IEEE International Conference on*, pages 361–368, nov. 2011.
11. A. Sirbu, A. Marconi, M. Pistore, H. Eberle, F. Leymann, and T. Unger. Dynamic composition of pervasive process fragments. In *Web Services (ICWS), 2011 IEEE International Conference on*, pages 73–80, july 2011.
12. H. Van Dyke Parunak. Interpreting digital pheromones as probability fields. In *Simulation Conference (WSC), Proceedings of the 2009 Winter*, pages 1059–1068, dec. 2009.
13. Ehtesham Zahoor, Olivier Perrin, and Claude Godart. Rule-based semi automatic web services composition. In *Proceedings of the 2009 Congress on Services - I*, pages 805–812, Washington, DC, USA, 2009. IEEE Computer Society.