Multiagent-based Adaptive Pervasive Service Architecture (MAPS)

Ray-Yuan Sheu, Michael Czajkowski, and Martin O. Hofmann
Lockheed Martin Advanced Technology Laboratories
3 Executive Campus, Suite 600
Cherry Hill, New Jersey 08002, USA
+1-856-792-{9057, 9792, 9711}
{rsheu, mczajkow, mhofmann}@atl.lmco.com

Greg Schow
Lockheed Martin Maritime Systems & Sensors
199 Borton Landing Road
Moorestown, New Jersey 08057, USA
+1-609-326-4114
gschow@lmco.com

ABSTRACT
In this paper, we describe a multiagent-based adaptive ad-hoc pervasive service architecture. It provides dynamic context-aware service and operation management including semantic service match-making, composition, execution, monitoring, and autonomic recovery. The intended application is decision support in tactical field operations. This domain requires dynamic adaptive composition of services, resources, and assets via ubiquitous and pervasive computing devices connected by mobile wireless peer-to-peer networks. Our goal is to provide on-demand pervasive services to meet users' dynamic situational information needs. Our solution uses distributed peer service composition agents to adaptively weave within and across three abstract service layers, i.e., the agent-based peer service composition agent layer, the agent-based wrapper service layer, and the resource and asset layer.

Categories and Subject Descriptors
I.2.11 [**Distributed Artificial Intelligence**]: Coherence and coordination, intelligent agents, structures, multi-agent systems.

General Terms
Algorithms, Design, Experimentation, Management, Performance, Reliability

Keywords
Multi-agent systems (MAS), peer-to-peer agents, service-oriented architecture (SOA), self-organized agents, ad-hoc dynamic service composition, mixed-initiative human and machine collaborations.

1. INTRODUCTION
Currently, many research efforts are focused on providing services for users with dynamic situational needs from a dynamically changing set of services. Meanwhile, there have been many advances in grid and cloud computing [1] [2] [3] [4] [5] and dynamic service composition [6] [7] [8]. A key challenge in military tactical field operations is the requirement for delivering services and promptly providing relevant information to users (e.g., tactical decision makers). Whether information is relevant depends on the situational context relative to the unit's current mission. Information is created by peers and by a variable set of assets (e.g., information provider systems, sensors, unmanned vehicles). Tactical field operations also constantly encounter challenges associated with environments where network communications, weather, terrain, and tactical conditions change frequently, which interfere with the delivery of services. Thus, tactical information service systems also need to be able to adapt to the surrounding environment conditions that affect the QoS (Quality of Service), e.g., when units move into built-up areas that limit the propagation of wireless communication signals.

We identify a number of technical solution concepts that promise to address the requirements of our target application, including dynamic service match-making with semantic inference, a dynamic networking service and node discovery from the underlying wireless networks, propagation mechanisms for situational contexts of users and services, automated semantic mapping and translation between new service domains and existing communities, partial matching for user requests, and scalable mechanisms for dynamic service discovery and matching. Semantic ontology alignment across service communities or domains necessary for semantic interoperability is also a recognized challenge for open domain commercial business applications. In addition, the computational complexity and scalability of service planning issues compounds the challenges on the above problems.

The behavior of tactical edge networks and computing environments are more unpredictable than most enterprise backbone networks. Consequently, the availability of pervasive computing services through dynamic composition and planning will face hard challenges. These challenges range from unexpected network data jitter, latency, and delay from seconds, minutes, hours, or days. The dynamic service composition solution will have to deal with recovery and resumption of service plans to overcome such unexpected random network behaviors.

We present a dynamic context-aware service-oriented computing architecture to overcome the unpredictable fast changing states of available resources, assets, and collaborators. In contrast, today's service oriented architecture (SOA) solutions do not yet address the issues of dynamic changes of services, resources, or networks. It is not practical to anticipate and implement a priori all possible matched services and compositions of workflows and plans to

AUPC'09, July 13-17, 2009, London, United Kingdom.

address emerging operational situations. We believe that most current service-composition approaches and architectures are not capable of supporting the morphing service demands in the uncertain environments with dynamic contexts around service consumers and providers in tactical operational environments.

Previously, we proposed an adaptive peer-to-peer agent-based approach (APPA) for dynamically deployed sensor networks [9] and an agent-based hierarchical task planning service composition approach using meta-planning concept [10]. It did not deal with the challenges of dynamic situations. There are several surveys of semantic service composition including the recent tutorial covering a comprehensive survey for system builders to develop semantic service composition systems [12]. This paper will focus on describing our proposed architecture approach by extending our previous works to support dynamically emerging service needs and capabilities in disadvantaged network communication environments encountered in tactical operations.

Dynamic situations and states of environments, users, and services are constantly changing in fast-tempo in dispersed low intensity operations. Situational context information will impact what services and operations are available at each moment caused by changing environments including networking, terrain, weather, etc. Also, tactical users will frequently have new situational needs due to the conditions in the field, available services and changes in operational mission needs. There exists a triangular dynamic relationship between changing environments, tactical users' needs, and dynamic services offered by tactical assets.

In tactical operations, it is important to acknowledge that human-in-the-loop for decision-making is a vital part of the collaborations. Human users will also play the roles of service or operation providers. It is critical to consider mixed-initiative man-machine collaborations among tactical users and assets. The user-agents will be used to serve the role of human proxies to participate in autonomic negotiations for collaborations including the initiations of users' situational needs. Human users must have roles in either the intermediate service selections or the final decisions on the agent-composed service plans.

Our proposed architecture will address the following key challenges to achieve the autonomic capability for meeting the fast-tempo situational needs of tactical users against dynamically changing available services.

2. RELATED WORK

We have identified the following research works related to our efforts:

- SOUPA Pervasive Ontology [21]: we extend SOUPA to support tactic operation specific extension such as mission, assets, services quality, and service effect while there are vast sets of ontology in tactic domains.
- Pervasive semantic service discovery and composition [11] [16] [17]. Our architecture supports both network-aware and agent-based service-aware discovery and composition.
- Distributed P2P replicated cache, e.g., DHT [20]: our architecture support different DHT mechanisms. We use the replicated registry approach due to the centralized server is suitable for P2P tactic operations.
- JXTA-SOAP and Web Service [22]: our architecture use agent abstraction layer to integrate with SOA web services and we intend to use JXTA-SOAP for agent-based service publishing, discovery to have agent-to-agent discovery using JXTA advertisement model in additional to our existing replicated cache approaches for service discovery.
- Self-Organizing multi-agents PMAS [23]: the MAPS architecture embraces the concept of self-organizing with its multiple abstract agent-based behavior controls. PMAS is a bio-inspired approach for self-organizing collective behavior that can be used as one of the core behavior engines.
- Multi-agent dynamic or distributed semantic service composition [12]. Our approach is dynamically ad-hoc grouping based upon network-aware and other grouping mechanism to provide dynamic services.
- Utility-based or market approach or utility-based approach for collaborations, negotiations, and coordination [14]. Currently, our architecture support extendable cost model with extendable attributes and algorithms for aggregations using weights, trust model such as the inference social networks for trust model [15].
- Situational context management [13] addressing the challenges of peer computing with scalability while considering the limitations of small computing devices and limited network communications and data quality.
- Cross-layer weaving of network, service, and semantic level of QoS management with dynamic situational contexts from users, assets, and the surrounding environments [18] [19]. Our approach is designed to integrate with agents in multiple service layers though Mokhtar's [24] also integrate networks.
- Flexible, efficient and scalable dynamic service match-making and hybrid matching algorithms to create hybrid QoS and utility evaluations such as OWLS-MX algorithms [18] [16] while considering partial matching for services.

3. APPROACH

3.1 Concepts

Based upon our previous work in P2P agents, HTN-based service composition, and distributed agents for software engineering, we propose the MAPS architecture (Figure 1) is to partition the problem space into three abstract layers, namely:

- The Physical Asset and Service Layer (PSL): the physical layer where actual assets, operations, existing services information servers reside.

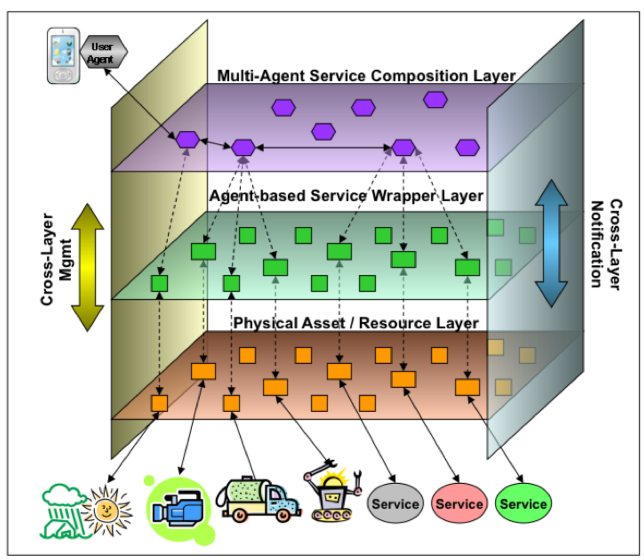

Figure 1. Multi-Agent Pervasive Service Architecture.

- The Agent-enabled Wrapper or SOA Service Layers (AWL): to provide SOA-enabled agents to wrap the existing assets or services using agent-based proxy wrapper to enable both agent or open SOA services presented in the network clouds.
- The Multi-Agent Service Composition Layer (ASL): to provide horizontal dynamic service compositions among the peer agents available in this layer. It is up to the service agents to dynamically find the matched services available on the AWL layer.

The advantages of the above three separate abstract layers are:

- Adaptive layers of abstraction of services: the adaptive three layers of hierarchical abstraction of services will provide manageable distributed computing for providing agent services, e.g., match-making against available sets of services. Hence, the computational complexity for service match-making is reduced accordingly. Every layer may have multiple service agents residing in different nodes that are capable of providing service match-making against proxy agents or assets inside its coverage. The combination of three hierarchical layers of service agents along with the horizontal coverage of available service agents will partition the computational search space of possibly matched service drastically. This makes the service match-making much more affordable in small computing devices.
- Isolation of concerns and dependency across layers: the service composition agents in the ASL will not have to concern themselves with the dynamically available SOA services and the states of the available services. The AWL layer agents will encapsulate the changes and the available assets and services in the PSL layer. The physical location and mobility of the available assets and services will not be visible across the service layer boundaries
- Leveraging of evolving network and SOA technologies: the evolution of hardware and web service technologies will not impact upper layers for dynamic service composition.
- Distribution of computation across peer nodes: The agents in both ASL and AWL layers can be deployed to any available nodes dynamically. In tactical operations, often a computer node will not be available. It is critical to have the capability to allow other peer nodes to resume the services or the plans of the failed nodes so that the tactical users can have continuity of services. Our APPA agents support this kind of dynamic migration of service agents.

4. ARCHITECTURE

4.1 Abstract Service Layers

4.1.1 The Multi-Agent Pervasive Service Layer

The key concept of the architecture is that it uses various kinds of distributed peer agents to dynamically initiate and collaborate to create end-to-end service plans through distributed peer match-making instead of centralized discovery and matching.

Peer agents in this dynamic ad-hoc community will collaborate to provide distributed resource sharing, service orchestration, and task allocations using a hybrid method combing semantic match-making and market utility approaches.

Typical SOA Web Service enterprise architecture will require some dedicated workflow application servers to serve as the execution engine, so-called workflow virtual machines such as the JEE (Java Enterprise) application server. This service layer provides server-less ad-hoc multi-agent service composition planning, execution, monitor, and recovery. Multiple types of agents are used to support the end-to-end services:

- Infrastructure Agents: to provide infrastructure service including agent registry, semantic service match-making, semantic inference service, and logic-based, semantic-based, or action-based planning for service composition.
- Mobile Agents: to provide dynamic distributed agent-based orchestration of services and executions while providing the options of migration from one node to another to recover failed services if comparable matched services are available. All the mobile agents have the same interfaces that other hosting infrastructure agents can use the same consistent interfaces, messages, and protocols.
- Execution and Monitor Agents: to provide service execution and monitoring of the generated service execution plan graph with dynamic states annotations. The Monitor agent will monitor the execution of each service and generate request for re-planning request if needed. These two agents are just special extension of infrastructure agents.

4.1.2 The Agent-based Service Wrapper Layer

This layer is designed to provide an agent-enabled wrapping service for SOA services including workflow management such as the BPELWS execution servers. We do not differentiate between atomic SOA services or composite ones. This layer consists of the following agents:

- Agent-based Wrapper (SOA) Service Layer: This layer is designed to provide an agent-enabled wrapping service for SOA services including workflow management such as the BPELWS execution servers. We do not differentiate between atomic SOA services or composite ones. This layer consists of the following:
- Proxy Agents: for SOA services or legacy ones and environment sensors. If a physical asset or service node does have an SOA-WS interface, a proxy agent will be used to wrap the web service to participate in cross-layer collaborations. However, if an asset of service node does not have a standard-based SOA web service and only has some kind of native interface, the agent will wrap the interfaces to provide agent-enabled services.
- Delegate Agents: This is a service agent provided match-making in a local community or group. The purpose of the delegate agent is to divide the search space of match-making services within the same grouping community.

4.1.3 Physical Asset and Resource Layer

This is the physical layer which providing the actual service or operations that might be hosted at remote locations. We provide two mechanisms for bridging the remote or physical assets. One is to use remote bridge wrapper agent do to the translation of messages and services. The other is to have local bridge agent if the remote asset is a closed computing environment with special access network.

4.2 Cross-Layer Management & Notification

In both circuit and IP switching networks, the robustness of its services is managed by centralized network management centers constantly monitoring the health and quality of service across the entire interconnected network and layers of protocols in routers and gateways. Lately, many wireless ad-hoc networks use the cross-layer aware approach to manage quality of service [28].

With a similar concept, the proposed MAPS architecture will provide support for cross-layer management and notification as part of context-aware situations from the three-layers, ASL, AWL, and PSL. The cross-layer notifications will inform the concerned agents on different or the same layers about the new situations and availability. For example, if an AWL agent detects a failure of a service, a notification can inform the service composition agent to find alternative compatible services to dynamically re-planning. Doing so, the tactical users will not have to be aware of the failure of his/her current services.

On the other hand, if a user requests a media service demanding certain network bandwidth service with authorization, the service agent ASL will seek available services on AWL layer. There could be an AWL agent wrapping the underlying network infrastructure. The cross-layer management will enable vertically the dynamic configuration of available networking service to provide temporary data path for the user.

Another usage of the cross-layer management and notification is to dynamically monitor application-specific service quality by some monitor agents on AWL. So, the summarized profile of service quality as a notification to the service composition agents on ASL layer to decide whether to dynamically re-plan the alternative services without the intervention from tactical users.

4.3 Dynamic Group Management

The architecture will operate service based upon the scope the group as a community to allow for peer agents to collaborate and create service composition for users. We identify three different group management mechanism and policy:

- Network-aware Group Management: this is the most primitive peer grouping mechanism as identified by the underlying network scope, e.g., different multicast address and ports, or different subnet within LAN or WAN. A network-aware policy will dictate how and what a peer agent can join the community. Similarly, the policy will also specify when a peer agent should join and leave the community. The important characteristic of the network-aware group management is that it provides the grouping over the physical networks. They could be either local or remote agents as long as there is a network path exists in between them.
- Logic Group Management: the grouping based upon some predefined echelons, interests, or needs to form a community. In tactical operations, this could be a small number of peer agents within the same or remote network-aware group. A peer agent will have satisfied the policies of both the network-aware groups it belongs to and the logic groups that the peer is going to join.
- Service-Aware Group Management: this grouping mechanism is a special extension of the logic group management. This grouping mechanism allows the agents to dynamically form a service-aware group to share the collaboration to achieve a common goal. After the goal is achieved, the group will be dissolved. For example, in tactical operations, some operators (via user proxy agents) with special skills might be temporarily joining a special assignment until the mission is completed.

4.4 Dynamic Context Management

Currently, in consumer mobile services, the location-aware context is the focus for mobile phone providers to enable dynamic services such as GPS tracking, nearby hotel and shops stores. Figure 2 illustrates the conceptual overview of our context-aware service composition model. In tactic field operations, there is

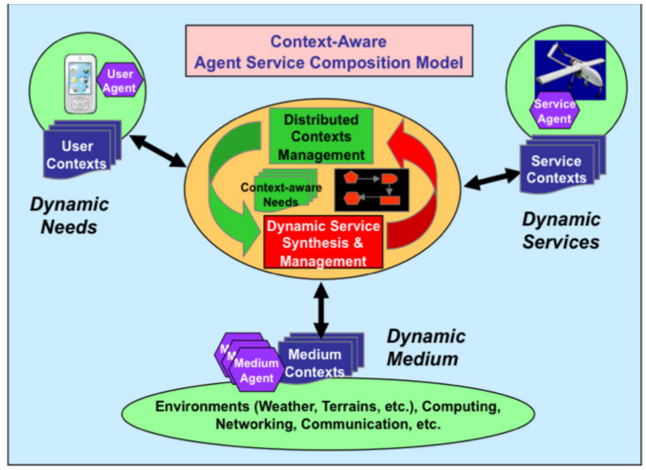

Figure 2. Context-aware service composition model.

much more situational context information that is important for missions in addition to the locations of tactical users. The tactical context information mostly will drastically change with fast-tempo. The need of efficient context management is critical for providing dynamic services. Certain context information will need to be broadcasted and others have to be either pulled or pushed depending upon specific situations. We identify that there are three different mechanisms to support by the architecture:

- Announcement mode service: a piece of context information will be broadcasted to all the agents within the same ad-hoc group or community. It could be initiated by either a peer agent, an external administrative agents, or tactical users. Examples are changes in environments, e.g., local weather forecast, policy updates such as group policy, network policy, domain-specific policy, e.g., tactical operation procedures. To avoid the typical information flood situation, the broadcast mode is limited to no relay from agents to agents. The architecture will detect whether an announcement is originated from the same group or not to avoid the cyclic announcement caused by relay of announcement. Currently, the architecture support this mode using its built-in group-wide distributed replicated cache (to be discussed further in next chapter) to minimize the protocol-based communications and reduce bandwidth usage.
- Push mode service: a peer agent detects some new context information and will publish updates to other peers based upon previous subscription of the topic by those peers. Or, if a group policy permits a proactive alerting of mission related information, the agent will publish the update to its peer agents about the new context updates. The architecture supports this with a persistent subscription mechanism built into the agent infrastructure. The peer-to-peer persistent subscriptions will survive either agent or node failures with autonomic resumptions until explicitly cancellation. This is an important feature for tactical users. For example, some blue force context-aware information of an asset might be concerned to a specific tactical user until the mission is over. The push mode is suitable for situational context that is only concerned by some peers within the same group. Only the interested peers should initiate the persistent subscription.
- Pull mode service: a peer agent will dynamically ask other authorized peers to provide context information regarding a specific topic. The pull-mode service is a protocol-based query service offered by all the participating peer agents.

This should only be used for dynamic specific needs of context information. For example, a peer agent is interested in the context and states of other peer agent's service and the type of context information is not setup by the group policy for either announcement mode or push mode. However, to avoid excessive usage and bandwidth consumption of such service, a policy can be used to limit what kind of context information is allowed to use this dynamic query service.

The architecture will need to support not only the propagation of context information but also the accumulative nature of information that could cause the stale information for tactical users.

For each context, it will have associated ontology context policies to describe its usage:

- Scope Policy: to specify how far the information should be delivered including relay of context across groups and receiving agents. The scope policy will be used to specify the scope of propagation:
    - The policy for context propagation model for either push and pull mode including which agents are allowed to receive context updates.
    - The context policy should specify whether relay service is allowed by the receiving agents. In addition, it is necessary to specify whether the receiving agents having multiple group memberships will be allowed to propagate the updates to other groups.
- Lifecycle Policy: to indicate how long the type of context information should be kept in some temporal ontology. So, the receiving agents can purge out the context information internally. This should be done using the lifecycle policy of context across all the peer agents for each grouping communities.
- Validity Policy: to describe how the validity of information should fade, e.g., linear fading, exponential decade fading, or one-or-zero fading. The fading of validity will be expired by the lifecycle property above. The expired piece of context information should be either removed or updated with new context information.
- Confidence Policy: to express the degree of confidence that an information producer agent believes to be accurate in numerical scale or fuzzy labels, e.g., high, low, medium, etc. This information is purely designed to help the recipient agent to decide how much confidence it can have for service composition or other decision-making services.

4.5 Hybrid Distributed Service Composition

We propose to use a hybrid approach to provide distributed service composition to address the challenges of tactical field operations such as, no dedicated servers, small computing devices, disadvantage networks, etc.

With the three abstract service layers as defined earlier, the distributed small computing resources will be able to handle much limited size of search space for performing service match-making.

We partition the distributed service composition into the following steps with reentrance allowed from any following step to provide dynamic re-planning to deal unexpected situational changes.

1. Initial service plan generation: our previous work used high-level task decomposition planning with hierarchical planner. However, it cannot handle the dynamic states and available service for alternate plans during the iterative steps of matching and discovery of services. The proposed architecture will support different algorithms such as OWLS-MX and other Golog-based or ADL-based service composition approaches. Currently, most of the known approaches for service composition utilize a centralized algorithm. The output of this step is to create an initial set of potentially matched services within the proposed service plans. At this moment, the service matched is only a nominal function-level match. For example, a fuel service agent may not have enough amount of fuel to provide as matched service. It doesn't mean that the selected service agent have the actual service quality available. This is the limitation of the existing OWL-S or any function-based service match-making.
2. Distributed Market-based (Utility-based) Dynamic Matching: the initial set of potentially matched services plan is created according to the OWL-S service specifications from the above step. This step will use the MAPS distributed auction method to negotiate with the "matched" service agents to estimate the abstract cost model for providing the "matched" service. The cost model is consisted of the cost ontology with constraints with abstract cost attributes and weight factors. The "coordinator" (auctioneer) agent will use the abstract model submitted by the potential candidate service agents to select the best matched services to compose the entire service plan for a user's service request. The auctioneer agent may further apply its own stochastic model of the learned trust model to combine with the auction bidding to decide the better matched services within the service plans. The output this step is to create the much limited candidate service plans to be presented to tactic users as the part of "human-in-the-loop" for decision-making.
3. Distributed Execution and Monitoring: once the originating user for a service request is presented with a set of potential candidate service plans. The user will make selection and ask the service execution agents in the ASL layer to start the execution through available execution and monitor agents. Each service plan will have one execution agent and one monitor agents in ASL. In the AWL layers, there will be multiple subordinate proxy agents working with the execution and monitoring agents. Conversely, a proxy agent in AWL layers may have one-to-many relations into the service composition agents in the ASL layer. The service plan will be used as the service monitor graph by extending and updating the service plan with actual execution status. When there is a deviation, the execution monitor will submit the failed service plan with the graph of execution states back to step 1 to do dynamic re-planning from the point the failure. Doing so, the service composition agents will not have to start from the scratch state to create service plans.
4. Execution completion and cleaning up: when the execution of a user's service plan is successfully executed, the notification will escalate from the PSL layer up onto the higher layers. During the escalating of the completion, the subordinate agents will clean up its working space to free up its internal resources. Finally, the user will be informed the success states with its execution of plan graph with states and dynamic re-planning history to users. This is the end of the four-step dynamic service composition in distributed peer-to-peer computing environments for each service request from user.

5. CONCLUSION, LIMITATION, AND FUTURE WORK

In this paper, we identify the challenges and needs of dynamic service match-making and service composition for meeting tactical field operation users need with ad-hoc wireless networks. We enumerate some of the issues in providing dynamic service composition and planning in distributed pervasive small computing devices.

We illustrate the concepts and capabilities of the proposed architecture to address the challenges. We do not address the semantic alignment across open domains of SOA services. Our efforts are in the relatively closed domain of tactical operations. Hence, our computational complexity will be relatively confined with respect to the search domains than solving the challenges in open domains with automating the service composition and workflow planning using semantic SOA service. However, tactical field operation presents a shifted set of challenges with the dynamicity in both user requirements with fast-tempo, the dynamic of the surrounding environments and the availability and states of potential matched services.

We are currently conducting experiments using different methods of service discovery, match-making, dynamic service planning to compare the performance and robustness. We need to further address the high-churning rate of node and available services dynamically while considering the scalability for future experiments especially for tactical field operations. Also, the multi-level of security of data and service sharing is not fully addressed in current implementation yet though policy management for dynamic filtering is supported in the MAPS.

6. REFERENCES

[1] BEinGRID (Business Experiments in Grid), sponsored by European Union under the Sixth Framework Programme (FP6), http://www.beingrid.eu/.

[2] UFORIA (EU Fusion for International Thermonuclear Experimental Reactor Applications), sponsored by European Union under the Seventh Framework Programme (FP7), http://www.euforia-project.eu/EUFORIA/.

[3] Enabling Grids for E-sciencE, http://egee2.eu-egee.org/.

[4] E-science grid facility for Europe and Latin America (EELA-2), http://www.eu-eela.eu/.

[5] EU DataGrid, http://eu-datagrid.web.cern.ch/eu-datagrid/.

[6] W. Zhu, Z. Du, and S. Zhu, “Dynamic Service Composition Based on Peer-to-Peer Network,” Service-Oriented System Engineering. The 2nd IEEE International Workshop Oct. 2006.

[7] M. Xu, J. Chen, Y. Peng, X. Mei, and C. Liu, “A Dynamic Semantic Association-Based Web Service Composition Method,” Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence, pg 666-672, 2006.

[8] A. Brogi, S. Corfini, and R. Popescu, “Semantics-Based Composition-Oriented Discovery of Web Services,” ACM Transactions on Internet Technology, Vol. 8, No. 4, Article 19, Sept. 2008.

[9] R.-Y. Sheu, M. Czajkowski,, and M.O. Hofmann, “Adaptive Peer-to-Peer Sensor Networks (APPA),” A workshop of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems. May 14-15, 2007, Honolulu, HI.

[10] M. Czajkowski, A.L. Buczak, and M.O. Hofmann, “Dynamic Agent Composition from Semantic Web Services,” LNCS, Semantic Web and Databases published by Springer Berlin / Heidelberg 2005.

[11] S. Kalasapur, M. Kumar, and B.A. Shirazi, “Dynamic Service Composition in Pervasive Computing,” IEEE Transactions on Parallel and Distributed Systems, Col. 18, No. 7, pp. 907-918, July 2007.

[12] K. Sycara, M. Paolucci, J. Soudry. and N. Srinivasan, “Dynamic Discovery and Coordination of Agent-based Aemantic Web Aervices,” IEEE Internet Computing, Vol. 8, Issue: 3. 2004.

[13] A. Sardinha, J. Rao, and N. Sadeh, “Enforcing Context-Sensitive Policies in Collaborative Business Environments,” Proc. of the First International Workshop on Security Technologies for Next Generation Collaborative Business Applications, Istanbul, Turkey, April 20, 2007.

[14] G. Lai, K. Sycara, and C. Li. A Decentralized Model for Automated Multi-attribute Negotiations with Incomplete Information and General Utility Functions. Journal of Multi Agent and Grid Systems, Vol. 4, No. 1, January 2008.

[15] U. Kuter and J. Golbeck, “Sunny: a new algorithm for trust inference in social networks, using probabilistic confidence models,” in Proc. of the 22nd National Conference on Artificial Intelligence, 2007.

[16] A.B. Hassine, S. Matsubara, and T. Ishida. “Dynamic Horizontal Composition for Semantic Web Services: An Investigation of Real Use,” Presented at the 6th International and 2nd Asian Semantic Web, 2007.

[17] S. Bianco, “Dynamic Service Composition in Pervasive Environment”, INRIA, http://www-rocq.inria.fr/who/Sebastien.Bianco/ReunionDesDevs_25_06_2008_Remaniee2.pdf.

[18] M. Klusch and B. Fries, “Hybrid OWL-S Services Retrieval with OWLS-MX: Benefits and Pitfalls,” SMR2, CEUR-WS, Vol. 243. 2007.

[19] J. C. Mundarath, P. Ramanathan, and B.D. Van Veen, “A Quality of Service Aware Cross-layer Approach for Wireless Ad Hoc Networks with Smart Antennas,” Ad Hoc Networks archive, Vol. 7, Issue 5, pg. 891-903, July 2004.

[20] H. Balakrishnan, M.F. Kaashoek, D. Karger, R. Morris, and I. Stoica, “Looking up Data in P2P Systems,” in Communications of the ACM, February 2003.

[21] H. Chen, F. Perich, T. Finin, and A. Joshi, “SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications,” International Conference on Mobile and Ubiquitous Systems: Networking and Services, August 2004, pp. 258-267.

[22] M. Amoretti, M.C. Laghi, F. Zanichelli, and G. Conte, “JXTA-SOAP: Implementing Service-Oriented Ubiquitous Computing Platforms for Ambient Assisted Living,” Proc. of the European Conference on Ambient Intelligence 2008, LNCS, Vol. 5355, pg. 75 - 90.

[23] M. Bakhouya and J. Gaber, “Self-organizing Approach for Emergent Multi-agent Structures,” CECCO’06, July 8-12, 2006, Seattle, WA.

[24] S.B. Mokhtar, A. Kaul, N. Georgantas, and V. Issarny, “Efficient Semantic Service Discovery,” in Pervasive Computing Environments, IFIP'06, LNCS 4290, pp. 240–259, 2006.