

A Performance of Monitoring Service Systems

¹Anil Kumar Dasari, ²P.Nageswara Rao

^{1,2,3}Dept. of CSE, NOVA College of Engineering & Tech., Jangareddygudem, AP, India

Abstract

We have presented a framework and an approach for Multilevel monitoring of service systems. It draws inspiration from the language-action perspective and extends recent work related to commitments. The key contributions, specific to the use of language-action perspective, include the following: Specification of an ontology of communicative acts that adds semantic content to messages. Support for the specification of policies that address standard and extended effects of communicative effects for multilevel monitoring. In addition, the framework specified supports the following Support for the specification of abstractions over agents and their operations, and decoupling operations from commitments via a mapping specification. . Service system specifications for an arbitrary number of services and processes. Specification of message semantics. Specification of local service behaviors that contribute to the participation in multiple conversations. Following the design-science approach the framework embodies our working theory for multilevel monitoring, which we evaluate by. Demonstrating feasibility application to scenarios. Comparison against prior efforts. The evaluation demonstrates comparative advantages of the framework against similar approaches, and how the capabilities of the framework address the needs of multilevel monitoring of service systems. Future work is focused on applying the framework to real-world case studies, and extending the approach to encompass additional levels suggested by LAP frameworks, such as contracts and multiple contracts that require ongoing relationships. For example, the goal hierarchy we have outlined contains two goal categories (agent service and agent protocol goals). We anticipate additional goal categories to account for complex service systems. The framework we have outlined provides an important foundation for these later investigations.

Keywords

Monitoring, Services, Processes, Language Action, Speech Acts

1. Introduction

In supply chains, e-commerce, and Web services, the participants negotiate contracts and enter into binding agreements with each other by agreeing on functional and quality metrics of the services they request and provide. The functionality of a service is the most important factor, especially for discovering services. Once discovered, however, services are engaged, composed, and executed by the participants' negotiating over QoS metrics to maximize their profits. Negotiation is a process by which agents communicate and compromise to reach agreement on matters of mutual interest, while maximizing their individual utilities. Negotiation for QoS-aware services is currently limited to primitive QoS verification methods or sorting and matching algorithms. We extend current techniques by presenting an optimal negotiation procedure that considers the cost to reach an agreement for QoS-aware service engagement and contracting. In general, negotiation is a technique for reaching mutually beneficial agreement among parties via communication. Negotiation in QoS-aware services involves a sequence of information exchanges between the parties to establish a formal agreement among them,

whereby one or more parties will provide services to one or more other parties. The agreement typically involves QoS issues [1]. By QoS, we refer to the non-functional properties of services, such as performance, cost, reliability, and security. To meet the requirements of service requestors, multiple issues, including both functional and non-functional, need to be taken into account during service advertisement, discovery, composition, and delivery. Preist [10] has discussed how negotiation plays an important role in reaching a service agreement for a semantic Web service. In this paper, we focus on the optimal strategy of efficiently negotiating multiple issues to reach an agreement that gives both a requestor and a provider their maximum utilities. Many researchers have investigated multiple-issue negotiation [3-4,7] . Fatima et al. [3] presented an optimal agenda and procedure for two-issue negotiation by introducing two negotiation procedures: issue-by-issue negotiation and package deal. For n-issue negotiation where $n > 2$, which is common in service-oriented negotiation, the computational cost to reach a package deal might exceed the benefits obtained by optimizing the participants' utilities. By considering both utility optimization and computational efficiency, we propose the coalition deal that is suitable for multiple-issue negotiation, especially in the case of multi-issue negotiation for services [2]. A typical real world service-oriented environment is dynamic, competitive, and partially observable. Semantic Web services, as envisioned by Berners-Lee, are intended to be applied not statically by developers, but dynamically by the services themselves through automatic and autonomous selection, composition, and execution. Dynamic selection and composition first require service requestors to discover service providers that satisfy the requestors' functional requirements. Second, the requestors and providers negotiate nonfunctional requirements (QoS), including cost and qualities such as response time, accuracy, and availability. As one step toward real-world service-oriented computing, many efforts have been made to automate service negotiation in a Web service environment. Current standards for Web services do not support automated QoS negotiations. As a result, several researchers have attempted to merge negotiation from the MAS domain into QoS-aware Web services. Ran [13] proposes to enrich current UDDI registries by extending the SOAP message format and the UDDI data structures to describe QoS information. Petrone [9] proposed a conversation model to enrich the communication and coordination capabilities of Web services by adapting agent-based concepts to the communications among Web services and users. Maximilien and Singh [6] propose a Web service agent framework (WSAF) with a QoS ontology. When a service consumer needs to use a service, WSAF will create a service agent that can capture a consumer's QoS preferences and select the most suitable service. Service negotiation involves both functional and non-functional issues. We cannot apply existing multiple-issue negotiation models to service negotiation and contracting directly, because existing models often make the limiting assumption that agents know the private information of their opponents, and their theoretic models do not take computational cost into consideration. Therefore, these models do not fit typical competitive environments, where self-interested agents engage in on-line QoS negotiation for services. In [4], agents know the incomplete preference information about their opponents and exploit this information to improve negotiation

efficiency. This work is thus limited to cooperative negotiation, where agents care about not only their own utilities, but also equity and social welfare, which is not the common case in Web service environments. [5] show the formation of a coalition is a dominant strategy under a proportionally fair divisible auction, by which agents bid for services and computational resources. A negotiation model is defined in [17] to allow agents to generate and evaluate proposals by employing a set of strategies and tactics within the model to reach an agreement for service provision. Sandholm and Lesser [16] discuss the issues in automated negotiation among bounded rational self-interested agents. In the context of task allocation negotiation, they present a negotiation protocol to support leveled commitment by introducing counter-proposals into the Contract Network Protocol. The outcome of multiple-issue negotiation depends on not only strategies, but also the procedure by which issues will be negotiated. Different procedures yield different outcomes. Based on an incomplete information assumption, Fatima et al. [3] discussed two procedures for multiple-issue negotiation: issue-by-issue and package deal. For two-issue negotiation, they determined the equilibrium strategy for these procedures and analyzed the optimal agenda and procedure. They concluded that the package deal is the procedure that provides agents with optimal utilities for two-issue negotiation. They did not address the computational cost with increasing issue size.

II. Existing System

The idea of conversations has been explored in the form of service interactions by a number of researchers [26-28]. With the exception of Moore [29], Singh and colleagues [28], and Umapathy and Purao [18], the use of communicative acts has, however, not been explicit. Further, the multilevel frameworks suggested by language action theorists [16-17] have not been used in this context other than the possibility suggested by Moore [29], and the exploration by Umapathy and Purao [18]. Our definition of service interaction protocols, therefore, builds on that offered by [30]: An interaction protocol is defined as the “rules of engagement” among interacting participants. It includes the possible set of actions that each may perform, and the order in which these must be performed. Concerns that need resolution for the design of interaction protocols, therefore, include: 1) specificity; 2) semantic content; and 3) compos ability. The first concern deals with specificity-abstraction dimension. For example, an abstract specification may represent interactions among services (e.g., racing condition [31]) that is domain-independent. On the other hand, a specific protocol, e.g., for shipment, may be domain-dependent.

The second concern is semantic content, i.e., the nature of the interactions. For example, a protocol specification may use just the request-reply mechanism, relying on message content (e.g., destination, payment info) to capture content. On the other hand, the message itself may capture the semantics to distinguish actions, such as direct, inform, cancel, and others. Finally, compos ability [31] is concerned with how a designer might specify the process by combining protocols [32]. To illustrate protocol specification, we draw on a common business process for purchasing, drawn from the Secure Electronic Transactions (SET) standard [33], originally described by Desai et al. [28]. Four main roles collaborate to carry out this process: 1. the Customer, who wants to purchase items, 2. the Merchant, who sells items, 3. the Shipper, who transports the physical items to the Customer, and 4. the Payment Gateway, who authorizes payments. Fig. 3 (adapted from [28]) illustrates how the four agents collaborate, denoted as circles. The rectangles at the role's

edge denote data transferred via labeled service messages. The numbers indicate typical sequencing. The ordering on multiple outward edges from a process (e.g., Merchant: task 9) is nondeterministic—the implementation can choose any ordering, including concurrent. Execution of the purchase process requires interaction protocols, which may be specified with message sequence diagrams (see Fig. 4). They describe how each message (communicative act) results in commitments (CC), which are described in the next section. Each sequence specifies a protocol (e.g., Payment, Shipment, and Order). A designer can compose these to design an operational system: Shipping protocol _ Payment protocol _ Order protocol _ Purchase process

A design mapping specifies how the protocols apply to the purchasing process. Part of that mapping is presented here (where x/Y means that x is substituted for variable Y):

III. Proposed Work

Accessing the EIP knowledge base requires identifying integration requirements from the BPMN models. One straightforward heuristic to do this is to identify interactions among participants in a business process [13]. An “interaction” defines a sequence of adjacent tasks that implicates two or more performers [14], including the roles ‘initiator’ and ‘responder’ [15]. This heuristic (change in performer) is useful for detecting and defining an ‘interaction’ in the business process. Speech acts can then be used to describe the actions performed; and the sequences of speech acts can be used to codify the interaction. Deciding on the appropriate speech act to describe the actions performed, however, requires yet another intermediate mapping. This mapping is provided by a characterization of tasks using “action types” [12].

Action types represent high-level business actions performed by participants through communication directed towards other participant(s) [16].

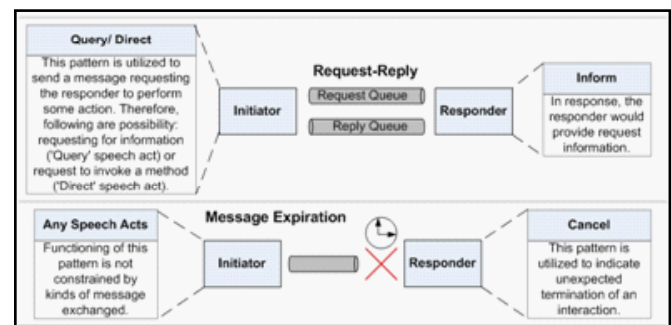


Fig. 1:

The complete sequence of mappings that allows access to EIPs based on integration requirements involves following activities:

- Identify interactions in the BPMN models
 - Characterize each task using action types
 - Map each task against possible speech acts
 - Identify EIP using sequence of speech acts
- The components that help realize this mapping include: (a) a set of action types that depict high-level business actions to categorize tasks in BPMN models, and (b) associations between action types and speech acts. We obtain the first, high-level business actions, to categorize tasks from business activity behaviors described in the Unified Modeling Language (UML) specification [17]. We construct the second by mapping the activities indicated in the action types against intended actions performed through the speech acts. Table 1 lists eleven action types that resulted from this analysis along with associations between action types and speech acts. The mappings informally described above are captured in a knowledge-base that implements the speech act based mechanism

to structure and access EIPs. It captures a number of concepts such as task, action type, speech act, EIP and relationships among EIPs [18]. We follow the definition of ontology simply as “an explicit specification of a conceptualization” [19]. The representation allows description of concepts and expresses relationships and constraints between concepts in a declarative formalism [20]. We developed ontological representations of the speech act based mechanism using Web Ontology Language (OWL) [21]. OWL provides a standardized semantic mark-up language for publishing and sharing ontologies intended to be processed by applications [21], and supplies the required formalisms to encode hierarchies of classes that describe concepts and relate these classes to each other using properties in a descriptive logic based constraint language [21-22]. The OWL knowledge base consists of five classes – Action Type, Pattern, Interaction, Speech Act, and Task; and nine properties – hasActionType, hasAssociation, hasSender, hasSenderSA, hasSpeechAct, hasReceiver, hasReceiverSA, hasPattern, and hasPerformer. Figure 2 provides conceptual model for the OWL knowledge base. Task class is used for capturing business tasks involved in integration requirements. Performer of Task individuals is stored in string data type hasPerformer property. Each Task individuals is related to maximum one Action Type class through hasActionType property. Action Type class consists of 11 sibling classes each capturing specific high-level business activities that represents actions performed by tasks depicted in BPMN models.

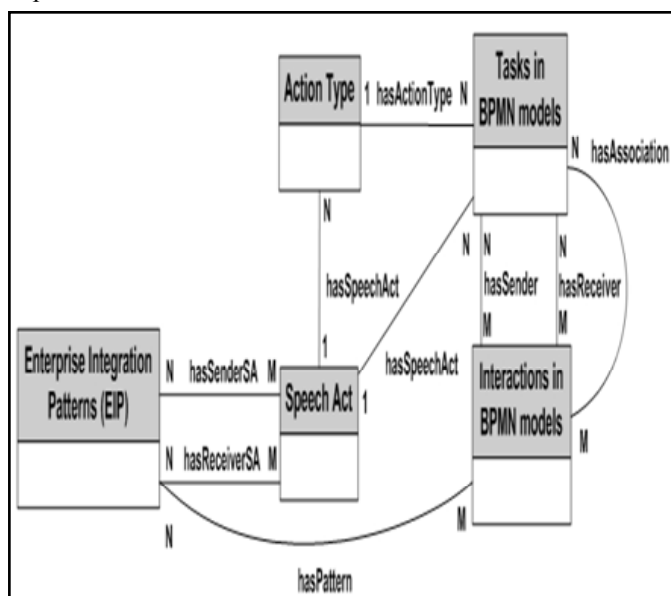


Fig. 2:

IV. Results

In order to implement the zealous and privacy preserving algorithm performance on basis on keywords, query, click and query pairs.

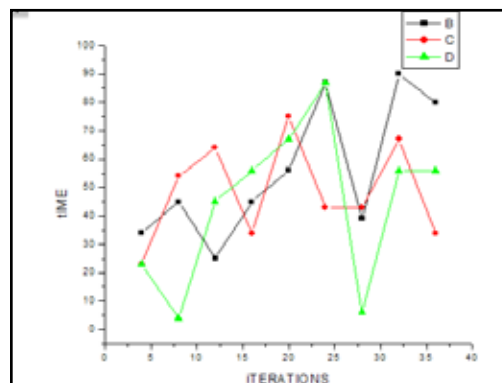


Fig. 1:

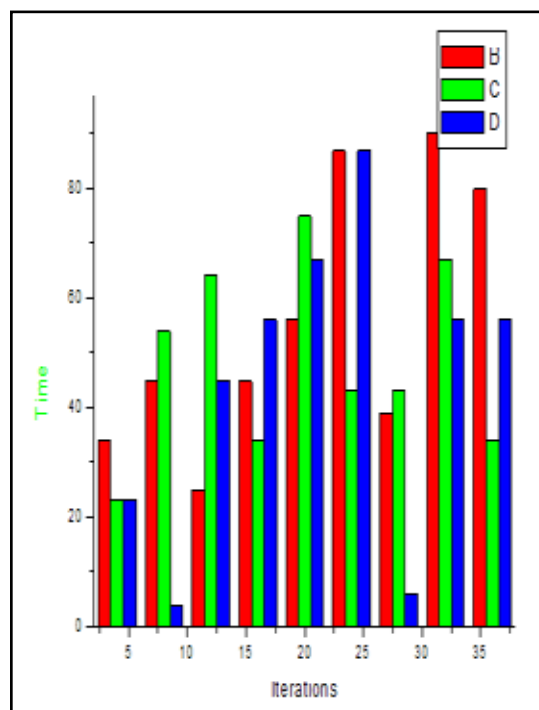


Fig. 2:

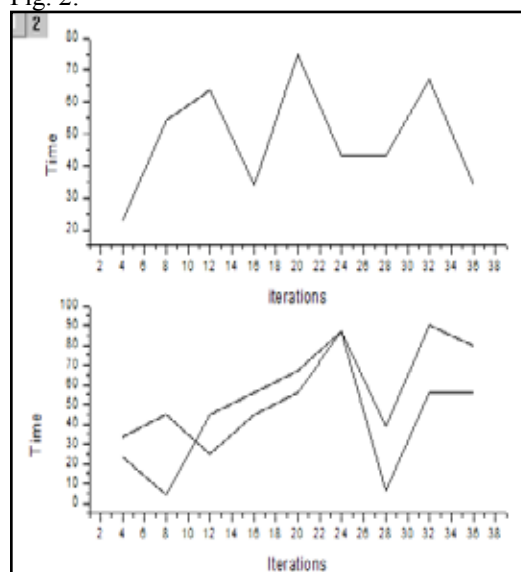


Fig. 3:

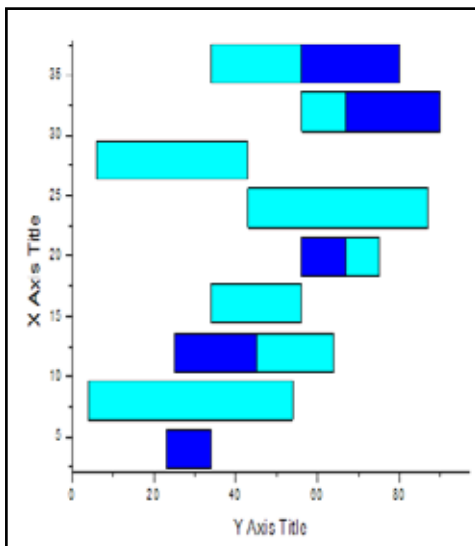


Fig. 4:

V. Conclusions

Business processes are increasingly distributed and open, making them prone to failure. Monitoring is, therefore, an important concern not only for the processes themselves but also for the services that comprise these processes. We present a framework for multilevel monitoring of these service systems. It formalizes interaction protocols, policies, and commitments that account for standard and extended effects following the language-action perspective, and allows specification of goals and monitors at varied abstraction levels. We demonstrate how the framework can be implemented and evaluate it with multiple scenarios that include specifying and monitoring open-service policy commitments.

References

- [1] Gartner-Group, "SOBAs Will Revolutionize Application Integration", [Online] Available: http://www.searchwebservices.techtarget.com/originalContent/0,289142,sid26_gci965822,00.html, 2002.
- [2] J. Dang and M.N. Huhns, "Coalition Deal Negotiation for Services," Proc. First Int'l Workshop Rational, Robust, and Secure Negotiations in Multi-Agent Systems (RRS '05), p. 67, 2005.
- [3] L. Brownsword et al., "System-of-Systems Navigator: An Approach for Managing System-of-Systems Interoperability," Technical Note CMU/SEI-2006-TN-019, <http://www.sei.cmu.edu/publications/documents/06.reports/06tn019.html>, Oct. 2006.
- [4] J. Lee, K. Siau, and S. Hong, "Enterprise Integration with ERP and EAI," Comm. ACM, vol. 46, no. 2, pp. 54-60, 2003.
- [5] D. Robey, D.L. Farrow, C. Franz, and R. Franz, "Group Process and Conflict in Systems Development," Management Science, vol. 35, no. 10, pp. 1172-1191, Oct. 1989.
- [6] C. Brown and I. Vessey, "Managing the Next Wave of Enterprise Systems: Leveraging Lessons from ERP," MIS Quarterly Executive, Vol. 2, No. 1, pp. 65-77, 2003.
- [7] M. Hellinger, S. Fingerhut, "Business Activity Monitoring: EAI Meets Data Warehousing," Business Integration J., 2002.
- [8] W.N. Robinson, "A Roadmap for Comprehensive Requirements Monitoring," Computer, pp. 64-72, Vol. 43, No. 5, May 2010.
- [9] D.K. Peters, D.L. Parnas, "Requirements-Based Monitors for Real-Time Systems", IEEE Trans. Software Eng., Vol. 28, No. 2, pp. 146-158, Feb. 2002.
- [10] A. Lazovik et al., "Associating Assertions with Business Processes and Monitoring Their Execution", Proc. Second Int'l Conf. Service Oriented Computing, pp. 94-104, 2004.
- [11] S. Turner et al., "Distributed Supply Chain Simulation Using High Level Architecture", Trans. Soc. of Computer Simulation, Vol. 18, No. 2, pp. 98-108, 2001.
- [12] F. Curbera et al., "Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI", IEEE Internet Computing, Vol. 6, No. 2, pp. 86-93, 2002.
- [13] J.L. Austin, "How to do Things with Words. Oxford, 1962.
- [14] J. Searle, Speech Acts: An Essay in the Philosophy of Language. Cambridge Univ., 1969.
- [15] J. Habermas, "The Theory of Communicative Action", pp. 465. Beacon, 1984.
- [16] M. Lind, G. Goldkuhl, "Generic Layered Patterns for Business Modelling", Proc. Int'l Working Conf. Language-Action Perspective on Comm. Modelling (LAP), 2001.
- [17] J.L.G. Dietz, "The Atoms, Molecules and Matter of Organizations", Proc. Int'l Working Conf. Language-Action Perspective on Comm. Modelling (LAP), 2002.
- [18] K. Umapathy, S. Purao, "A Theoretical Investigation of the Emerging Standards for Web Services", Information Systems Frontiers, Special Issue of Information Systems Frontier (ISF)—From Web Services to Services Computing: Technologies and Applications, Vol. 9, pp. 119-134, 2006.
- [19] C. Peltz, "Web Services Orchestration—A Review of Emerging Technologies, Tools, and Standards", technical paper, Hewlett Packard Labs, 2003.
- [20] J.E. Hanson et al., "Conversation Support for Business Process Integration", Proc. IEEE Int'l Enterprise Distributed Object Computing Conf. (EDOC), pp. 65-74, 2002.
- [21] J.E. Hanson et al., "Conversation-Enabled Web Services for Agents and e-Business", Proc. Int'l Conf. Internet Computing (IC), pp. 791-796, 2002.
- [22] L. Polanyi, "A Formal Model of the Structure of Discourse", J. Pragmatics, Vol. 12, pp. 601-638, 1988.
- [23] X. Fan et al., "A Theoretical Framework for Proactive Information Exchange in Agent Teamwork", Artificial Intelligence, Vol. 169, pp. 23-97, 2005.
- [24] P. Stone, M. Veloso, "Task Decomposition, Dynamic Role Assignment", and Low-Bandwidth Communication for Real-Time Strategic Teamwork", Artificial Intelligence, Vol. 110, pp. 241-273, 1991.
- [25] G. Hohpe, B. Woolf, "Enterprise Integration Patterns: Designing, Building and Deploying Messaging Solutions. Addison-Wesley, 2003.
- [26] G. Decker et al., "Formalizing Service Interactions", Proc. Fourth Int'l Conf. Business Process Management (BPM '06), pp. 414-419, 2006.
- [27] A. Barros et al., "Service Interaction Patterns", Proc. Third Int'l Conf. Business Process Management, pp. 302-318, 2005.
- [28] N. Desai et al., "Interaction Protocols as Design Abstractions for Business Processes", IEEE Trans. Software Eng., Vol. 31, No. 12, pp. 1015-1027, Dec. 2005.
- [29] S.A. Moore, "On Conversation Policies and the Need for Exception", Proc. Autonomous Agents Special Workshop Conversation Policies, pp. 19-28, 1999.

- [30] K. Umapathy, S. Purao, "Exploring Alternatives for Representing and Accessing Design Knowledge about Enterprise Integration", Proc. 26th Int'l Conf. Conceptual Modeling (ER '07), 2007.
- [31] G. Decker, F. Puhlmann, "Formalizing Service Interactions", Proc. Fourth Int'l Conf. Business Process Management (BPM '06), 2006.
- [32] S. Purao et al., "Improving Reuse-Based Design: Augmenting Analysis Patterns Reuse with Learning", Information Systems Research, Vol. 14, No. 3, pp. 269-290, 2003.
- [33] SET, "Secure Electronic Transactions (SET) Specifications".



Mr. D. Anil Kumar is a student of NOVA college of Engineering & Technology, Jangareddy gudem. Presently he is pursuing his M.TECH from this college and He received his B.Tech from Acharya Nagarjuna University In the year 2010.



P. Nageswara Rao obtained his B.E , M.TECH from JNTU. Now he is working as Assoc. Prof in NOVA COLLEGE OF ENGG AND TECH. He has total 10 years experience and he worked in the VS Lakshmi college in kakinada Chaitanya college in kakinada and 2 years experience in NOVA college.