RESEARCH STATEMENT

John Harney (jfh@cs.uga.edu)

Broadly speaking, my research interests lie in applying theoretical artificial intelligence concepts, statistical methods, and elements of the semantic web to resolve problems in service-oriented computing systems. Service-oriented computing (SOC) offers a powerful approach to the assembly of complex distributed systems from independently developed software componenets. It is a vast and enormously complex discipline in computer science, and is increasingly finding utility in business, government, and science. As SOC is still in its infancy, it presents a wonderfully exciting and unique opportunity for the modern researcher to solve practical, real-world problems using concepts from different computing disciplines.

Specifically, my dissertation research has focused on maintaining optimal Web service compositions that execute in volatile environments. Web services (WSs) – platform-independent applications which can be published, discovered, and invoked over the Web – are fast becoming the building blocks of workflows utilized by business organizations. Subsequently, dynamic Web service compositions (WSCs) – automated workflows with WSs as their functioning components – are providing a promising direction for on-demand business requirements. These compositions often function in volatile environments, where the parameters of the participating services change during execution. Failure to adapt to such changes may result in a sub-optimal composition. Surprisingly, this problem is often viewed as an afterthought in the traditional services literature. I believe there is ample opportunity to improve performance simply by identifying changes in WSC environments and reacting to them accordingly. To this end, I have implemented various adaptation algorithms and techniques that optimize WSCs in the presence of such changes by borrowing from core concepts from AI (e.g. decision-theoretic planning, utility theory, and information theory just to name a few), statistics, and machine learning. I briefly describe my work below.

Background and Past Research

In the past few years, I have focused on adapting WSCs to environments that exhibit *data volatility*. Data volatility encompasses changes in the non-functional, or quality-of-service (QoS), properties of component Web services that may occur throughout the WSC's lifecycle. For example, a product may go out of stock affecting the availability of a supplier WS in a supply chain process, the network bandwidth may fluctuate affecting a WS response time in a real-time application, or the cost of a WS used for booking an airline flight may increase. The optimality of the WSC is dependent on the accuracy with which the composition model captures the environment, so that the composition may make the proper adjustments (such as replacing a component WS suddenly offering poor QoS with another). Thus, to remain optimal and cost-efficient in the presence of these changes, the WSC should possess up-to-date knowledge of the revised information during execution. To obtain this knowledge, an adaptive WSC should *query* component services for their revised parameter values. The revised parameters of the queried services may then be integrated into the model so that the composition may become optimal.

Querying for component services parameters, however, comes with its own attendant challenges. While revised information in some services may cause changes in the overall WSC, changes to other services parameters may have little or no impact. Additionally, WSCs typically operate over an open and large-scale system (i.e. the Internet), often making revised information queries tedious, time consuming, and costly. The process of querying must be carefully managed.

In my research, I have identified the desirable properties that a query management strategy for adapting WSCs should possess. Specifically, a querying method should:

- **Determine if the revised parameter information of a service is *useful*.** Not all changes will affect the composition. We should identify those services whose parameter changes will substantially impact the composition, so that we do not waste critical time and effort on unnecessary queries.
- **Determine if querying a service is *cost-efficient*.** The potential cost savings realized by obtaining the new information should outweigh the cost of querying.
- **Be *computationally tractable*.** It is essential that the querying method does not create a potential computational bottleneck such that exhorbitant resources are consumed and make the method impractical to use.

I have developed a sophisticated querying technique that addresses the challenges listed above.

Selective Querying Using the Value of Changed Information

To address the first two challenges stated above, I introduced a selective querying scheme that will suggest a query only when the queried information is expected to be suffciently valuable. My method, which I have called the *Value of Changed Information* (VOC) [1] query method, draws inspiration from an important concept in the AI literature called the Value of Perfect Information (VPI). VPI determines the amount of resources (e.g. time, money, etc) a decision making agent should be willing to pay for information prior to making a decision.

VOC identifies the best services from which to elicit revised information in a changing environment. Essentially, VOC is a measure of how badly, on average, a composition will perform as services change in the environment. The VOC-driven querying approach is myopic in that only one service is considered for querying at a time. For each candidate service in the composition, a VOC value is found by obtaining the value lost by continuing to utilize the composition as is currently being executed opposed to using the most optimal composition possible in the as that particular service changes. The largest of the VOC values (referred to as VOC^*) is then compared to the cost of querying for the revised information for that particular service. If the VOC^* is found to be greater than the cost of querying for the information for a particular service, then a query is issued to that service, the revised parameter information is integrated into the model, and the WSC is recomposed; the composition subsequently continues execution. Otherwise we issue no query, as adaptation is seen as to be too costly, and continue execution of the WSC as is. I demonstrated that using a VOC-driven querying strategy for adaptation resulted in a substantial reduction of overall average cost for the composition when compared to utilizing simple querying-driven adaptation heuristics. I have also shown that the VOC idea can be implemented on top of WSCs containing a wide variety of complex component service configurations and workflow patterns [4].

Mitigating the Computational Complexity of VOC-driven Selective Querying

The cost savings brought by using the VOC-driven selective querying strategy comes at a computational price. Extra computation is needed to for compositions to compute VOC values and issue queries. Such computational lag can be detrimental to dynamic, on-demand compositions, as it may consume a large quantity of computing resources as well as add time to WSC execution, rendering the VOC strategy impractical to use. Thus, I constructed two techniques that mitigate computational complexity that accompanies finding VOC. First, I alleviated the complexity of finding the VOC for individual services by observing that for particular values of the parameters, the WSC remains unchanged [3]. Such revised values may be ignored, as they do not change the WSC. With the aid of the gradient descent technique, I provided a simple and quick way to ascertain these values. Second, I reduced the time required to compute the VOC of the candidate service with VOC^*. Here, I utilized the expiration times often associated by service providers through a service level agreement (SLA) to the parameters of their services. We use the intuition that we need not consider querying those services for revised information whose previously obtained information has not expired. I incorporated this insight into the VOC formulation, and call the approach, the value of changed information with expiration times (VOC^{ϵ}) [2]. Because VOC^{ϵ} focuses the computations on only those services whose parameters could have changed, it is computationally more effcient than the traditional VOC. In implementing these techniques, I was able to demonstrate (both theoretically and empirically) a substantial drop in computational overhead in utilizing the advantageous VOC-based querying adaptation method.

Current Projects and Research

I am currently involved in two separate projects. First, I am in the process of building an extension of an eclipse plugin tool developed here at the University of Georgia called HALEY [5]. HALEY is an automated, scalable, end-to-end semantic Web service composition framework. My new plug-in will essentially act as a recommender system for HALEY, using the VOC-based querying method to identify services that WSC should query so that it may intelligently adapt to volatile environments. Second, I am investigating how the adaptation problem may be influenced when compositions employ *risk-sensitive* end user (i.e. process designer) objectives. Risk-sensitive preferences are an important type of preference structure that influence how process designers make high-stake decisions. This is true in many real world business processes that may be subject to huge possible gains or losses in resources (e.g. time, money, people, etc). As a result, we must adjust our previous adaptation scheme to consider risk.

Conclusions and Future Work

In the future, I plan to apply the principles I have learned in my dissertation research to projects on a grander scale. With the recent evolution of the programmable Web and Web 2.0 paradigms, the demand for dynamically adapting applications to volatility over the Web has never been higher. This can especially be seen in context-aware applications, where the performance of the application is highly dependent on the accuracy of environment sensing, and cloud computing, where resources must be properly managed based on the constant fluctuation of end user demand. Ideally, I would like to integrate the concepts I have established with other concepts (such as aspects of control theory) to develop sophisticated monitoring systems for these types of dynamic applications. Furthermore, while the topics of my current research have significant potential of further exploration, I am also interested in exploring and learning new areas in computer science in the future. Overall, I am looking forward to continuing my research career and learning from the brightest minds in business, industry and academia.

References

[1] J. Harney, P. Doshi, "Adaptive Web Processes Using Value of Changed Information", *International Conference on Service-Oriented Computing (ICSOC)*, pages 179–190, 2006

[2] J. Harney, P. Doshi, "Speeding Up Adaptation of Web Service Compositions Using Expiration Times", *World Wide Web Conference (WWW)*, pages 1023–1032, 2007

[3] J. Harney, P. Doshi, "Selective Querying For Adapting Web Service Compositions Using the Value of Changed Information", *IEEE Transactions on Services Computing*, Vol. 1, No. 3, pages 169–185, 2009

[4] J. Harney, P. Doshi, "Selective Querying for Adapting Hierarchical Web Service Compositions Using Aggregate Volatility", *IEEE International Conference on Web Services (IWCS)*, pages 43–50, 2009

[5] H. Zhao, P. Doshi, "Haley: A Hierarchical Framework for Logical Composition of Web Services", *IEEE International Conference on Web Services (IWCS)*, pages 312-319, 2007