Computer Science and Information Technology 3(4): 91-104, 2015
DOI: 10.13189/csit.2015.030401 http://www.hrpub.org

A Hierarchical Multilayer Service Composition Model for Global Virtual Organizations

Abiud Wakhanu Mulongo*, Elisha T. Opiyo Omulo, William Okello Odongo

School of Computing and Informatics, University of Nairobi, Kenya

Abstract A major benefit of service composition is the ability to support agile global collaborative virtual organizations. However, being global in nature, collaborative virtual organizations can have several virtual industry clusters (VIC), where each VIC has hundreds to thousands of virtual enterprises that provide functionally similar services exposed as web services. These web services can be differentiated on a high dimensionality of quality of service attributes. The dilemma the virtual enterprise broker is faced with is how to dynamically select the best combination of component services to fulfill a complex consumer need within the shortest time possible. This composite service selection problem remains a Multi-Criteria Decision Making (MCDM) NP hard problem. Although existing MCDM methods based on local planning are linearly scalable for large problems, they lack capabilities to express critical intertask constraints that are practically relevant to service consumers. MCDM global planning methods on the other hand suffer exponential state space explosion making them severely limited for large problems of industrial relevance. This paper proposes HMSCM: Hierarchical Multi-Layer Service Composition Model. HMSCM is based on the theory of Layering as Optimization Decomposition [28-31]. We view the service selection process as a "two layer network" where each layer is a subproblem to be solved. The objective of one of the layers is to maximize a local utility function over a subset of web service QoS attributes from a service consumer perspective. The objective of the other layer is to maximize a local utility function over another subset of web service QoS attributes from the perspective of the Virtual enterprise broker. We develop the algorithm: Service Layered Utility Maximization (SLUM) that extends the Mixed Integer programming model in [9]. We then formulate the problem at each layer in form of SLUM. Together, the two layers attempt to achieve the global optimization objective of the network. We show analytically how HMSCM overcomes the shortcomings of existing local planning and global planning service selection methods while retaining the strengths from each. i.e HMSCM is able to scale linearly with increasing number of QoS variables and number of web services while being able to enforce global intertask constraints.

Keywords Service Composition, Global Virtual Organizations, Hierarchical, Multilayer, Decomposition, Layering, Optimization, Mixed Integer Programming, Service Layered Utility Maximization

1. Introduction

1.1. Background and Context

Global virtual organizations are increasingly relying on service oriented architecture (SOA) as an information technology framework to quickly create value added business services from simple loosely coupled distributed services. Studies such as those in [2], [3],[4] confirm this claim. From the existing simple services owned by geographically sparse enterprises, a virtual enterprise broker [1] can quickly setup a complex service that meets complex consumer needs that cannot be satisfied by any one of the simple services. On the other hand, through SOA, each of the virtual enterprise participating within the consortium of global virtual firms has the chance to be discovered and selected to contribute in the provision of a composite service. This form of business agility facilitated by SOA is made possible through the concept of *service composition*. Service composition involves combining many services to produce a high value added composite service capable of fulfilling complex consumer request that cannot be fulfilled by any single service provider [5]. Therefore, in the context of virtual organizations and virtual enterprises, service composition reduces the time required to react to an external time varying consumer demands by promoting reuse of existing services owned by different enterprises within the virtual organization. This degree of agility is critical if both virtual enterprise brokers and virtual enterprises were to remain relevant in globally competitive market.

However, when applied to dynamical environments such as global virtual organizations, service composition is a difficult research problem that perennially remains open. How to efficiently select the best composite service to meet

complex service consumer needs is just one of the overarching problems in service composition that is under active research. This paper addresses the performance inefficiency issues that constantly plague the service composition process within complex global virtual business settings. The performance inefficiencies in service composition arise from the coupling of a myriad of factors. The most eminent issues are:

1. *The Large Scale of Services*

In a global virtual organization operating within a particular business domain, there are potentially hundreds to thousands of small to medium virtual enterprises offering competing functionally similar simple services. The total number of service providers summed from each category of services is even larger. Although in each cluster, the services may be functionally similar, they may be differentiated on some quality of service (QoS) criteria. Even when the differentiating factor is a single QoS parameter, the sheer numbers of services make the selection of the best composite service a challenge. To put this into perspective, consider a composite travel reservation product that contains four simple services: flight service, hotel service, insurance package and a taxi service. Assume further that for each of the simple services, there are 10 service providers. When a virtual enterprise broker is faced with a customer request enquiring for a trip, the VEB is required to select the best combination of four services, 1 from a pool of 10 candidate services. It's easy to show that there are 10^4 or 10,000 possible composite services from which to select the best service. A marginal change from 10 to 20 services per category exponentially escalates the solution space to 160,000 and 100000000 for 100 services per task. Algorithms that linearly scale with change in number of candidate services despite exponential growth in solution space are sought.

2. *High Dimensionality of QoS Decision Variables*

At the technology implementation layer, web services technology is the most widely used technology in realizing business services (in our case *virtual enterprise services*). Functionally equivalent web services (each web service provided by a different enterprise) can exhibit significant variations in quality of service along dozens of QoS parameters. A close examination of the number of papers on web service QoS such as [6], [7],[8],[9] reveal a wide range of important QoS parameters associated with web services.

The combination of the dimensionality of QoS attributes with even a small number of services exponentially increases the combinatorial complexity of the service selection problem. Intuitively the problem is expected to worsen as the both the number of QoS attributes and the number of candidate services grows larger. The challenge to the virtual enterprise broker transforms from just *how to select the best composite service from a large set services based on a single criterion* to *how to efficiently select the best combination service* from a huge *set of services on multiple criteria*. Further, in this case, the selection should factor in constraints and preferences that are either explicitly stated by the service consumer or implied by user needs.

The interaction of the above two issues makes the service selection problem a Multi-Criteria Knapsack NP hard problem.

1.2 The Issues

As pointed out earlier, service composition affords both virtual brokers and virtual enterprises the agility required to survive in a global virtual market dominated by cut throat competition.

However, as is evident too, the flexibility provided by service composition comes at a heavy cost – intensive time consuming computations. The high dimensionality of the variables and constraints to be considered coupled with the large scale of services makes the selection of composite services to remain a Multi-Criteria Decision Making *NP* hard Knapsack problem [10],[11],[12]. This should be significantly worrying especially to virtual brokerage firms because:

- From a service consumer's view point response time is the most critical performance parameter. Empirical evidence shown in [13],[14],[15], [16] and [17] , all lead to the same conclusion that service or software application response time or performance efficiency in general has the potential to attract or retain customers ; therefore it has the ability to cause significant gain or loss of business revenue. Specifically, according to these studies, 0.1 seconds is considered by the user as instantaneous response, 2 seconds as the tolerable waiting time and anything beyond 10 seconds as annoying. As noted in [16] these usability results are valid for all families of software systems and hence service oriented applications are not escapable.

So how to dynamically compose services that best satisfy every service consumer's current needs efficiently remains a worthwhile research problem albeit a difficult one

1.3. State of the Art Multi-Criteria Service Selection Strategies

Existing solutions to the problem stated in 1.2 follow either local planning or global planning strategies. Both strategies are based on the Multiple Criteria Decision Making (MCDM) [18] method .The objective in both methods is to maximize some utility function over a set of decision variables that are constrained. The utilities are computed using the Simple Additive Weighting [18] model. The current formulation of the local planning strategy works as follows. For each workflow task, identify the set of candidate services. Then compute the utility of each service over the set of web service QoS variables and select the service with the highest utility subject to some constraints.

The loose assumption here is that selecting a service with the highest utility from each service class locally, aims at global optimality of the resultant composite service. Techniques following local planning have recorded impressive linear scalability results on embarrassingly large service composition problems involving thousands of services. Furthermore, local planning may be the only practical optimization technique applicable depending on the objective at hand. For instance, some QoS decision variables may only be specific to a specific class of services based on the type of task making it impossible to compare constraints on these variables across tasks. However, a major setback of local planning is its inability to capture intertask constraints on decision variables shared across workflow tasks [9]. This limitation does not end here- service consumers are denied the opportunity to express critical constraints such as: the total service execution (or access) cost should not exceed a particular budget. Similarly it's infeasible to enforce constraints like the total execution duration of tasks should be less than some threshold value. Lastly, due to inability to express global constraints, local planning methods are only suboptimal. Being suboptimal should not be a big concern though. This is so because for large scale problems of industrial relevance, often suboptimal but more efficient solutions are sought [19],[20]. Perhaps the question should be whether or not local planning solutions converge to global optimality. This is beyond the scope of this paper.

Global planning based algorithms on the other hand overcome the limitations of local planning models by considering global constraints on workflow tasks. Given sufficient time, global planning is guaranteed to yield an optimal solution. Unfortunately, as demonstrated by Benatallah in [9], global planning methods severely suffer exponential state space explosion for large problems hence obtaining an optimal solution is computationally intractable. For instance the naïve global planning approach uses exhaustive search where it requires comparing generating m^n candidate services and computing utilities for each where m are the number of candidate services per task for n tasks. An alternative to exhaustive global planning is to apply Mixed Integer Programming, MIP [21] for optimization of composite service selection. MIP is an efficient technique for many optimization problems in which some variables take on integer values while other variables are continuous [22]. In web service selection, Benatallah et al in [9] goes ahead to provide an alternative global planning formulation that is based on MIP. The author empirically shows improved performance results on MIP over the exhaustive search. However, the author notes that MIP is still susceptible to exponential state space explosion and thus still limited to small scale service composition problems. This observation is also made in [20].

Another way of solving complex service composition problems is to cast the problem as a Satisfiability (SAT) Problem. In SAT, a problem is specified in form of propositional logic and derivative modelling formalisms such as Descriptive Disjunctive Logics (DDL). Although SAT problems are NP complete [23], many very efficient SAT algorithms exist today such as SATPlan [24], WalkSAT [25], and GraphPlan [26]. These algorithms are applicable to a large spectrum of practical problems. For instance within service composition research, SATPlan and SATPlan are recommended for complex operator large scale service selection [20]. Other closely related service selection optimization algorithms include A* and its variants, generic algorithms, Answer Set Programming (ASP). ASP is based on DDL and has been proven to be very efficient as exemplified by the work by Abert Rainer [27]. However, as a downside, SAT and other AI planning based approaches to service composition are limited in their scope of application in the following ways: - First, for most complex problems, it's always difficult to model them efficiently as SAT problems [22]. Second, AI planning and SAT solutions are more naturally suited to semantic web services composition. The reason for this is because; semantic web services are semantically annotated using AI like languages easily allowing for automated reasoning. But to date, semantic web service composition is yet to bear any fruits in commercial use. On the contrary, workflow based service composition based on WSDL services continue to enjoy strong industry support as they permeate many business applications. Third, generally, SAT and Constraint Satisfiability Problems are plagued by the same inadequacy seen in mathematical programming techniques such as MIP- the plague is exponential state space explosion [19]

As a last resort, one would aim for general purpose off the shelf mathematical programming and constraint programming solvers. However, general optimization packages are too generic to suit the specificities of different application contexts.

The gaps identified and summarized in section 1.4 obviate the serious need for more efficient service selection models that are industrially applicable.

1.4. Summary of the Gaps in the State of the Art

From the foregoing discussions it's irrefutable that existing methods for optimal service selection suffer a combination of the following issues:-

I. Inability to address critical optimization concerns such as the ability to express global constraints on tasks as exemplified as observed in local planning approaches.

II. Service consumers are required to specify their preferences by supplying weight values for all the set of available web service QoS parameters. When the dimension of such variables is large, it not only becomes too tedious for the user but the weight assignment process becomes less objective. For example it's too tempting to ask the end user to specify relative weights on QoS attributes like throughput, reliability and availability etc ,first because any Internet user would always expect that their service request is going to be successfully responded (100% expected reliability), by implication 100% expected

availability . Secondly, even if hypothetically, users were willing to trade off reliability or availability for instance, the nuances of these technical QoS terminologies can be too blurry to an end user for them to objectively assign relative weights accordingly.

III. Severely suffer from exponential combinatorial and state space explosion making them infeasible for ultra-low latency real-time industrial scale service based applications; the case of MIP, SAT and CSP.

1.5. Purpose of this Study

The main goal of this paper is to develop a more efficient composite service selection strategy which scales with the dimensionality of web service QoS decisions variables and increasing size of candidate web services without:-

- Sacrificing the ability of service consumers to express critical constraints spanning workflow tasks.
- Necessarily overburdening users to specify weight preferences on all web service QoS parameters.

If this goal is achieved then the first three gaps in section 1.4 will be filled.

1.6. An Overview of Our Approach

Towards this goal stated in 1.6, we propose a multi-layer service composition model dubbed *HMSCM* for *hierarchical multi-layer service composition model.* From an algorithmic perspective, HMSCM extends the MIP model originally formulated in [9] which is the basis for present service selection models that are based on MIP. Our departure from the current philosophy and practice is our fundamental rethinking about the structure of the service selection within the service composition problem. Instead of viewing composite service selection as one monolithic complex problem as it's the case today, we view it as a "network with multiple layers" in which each layer is a subproblem with a distinct objective function to be solved.

Contrary to the norm where utility maximization in web service selection is either entirely from a service consumer perspective or entirely from an end user perspective, here, our model supports the simultaneous maximization of both the end user utility and the utility of service provider (the virtual enterprise broker in this case). This leads to two optimization objective functions to be solved in a coordinated manner as opposed to a single objective function as is the case with all current approaches. Therefore in HMSCM, one layer strives to maximize the local utility from the point of view of the service provider (virtual enterprise broker) and the other layer trying to maximize the local utility from the point of the service consumer. We show that together, the two layers attempt to solve the global optimization objective. This (architectural) thinking is inspired by the formal theory of *layering as optimization decomposition* as described in [28], [29],[30],[31]. This theory is one of its kind that provides a framework for rigorous and formal design and analysis of layered communication architectures. The theory has led to the modularized and distributed reformulation of the Network Utility Maximization problem [32]. The reformulation of the NUM problem based on the theory has been applied to re-engineer the TCP/IP protocol stack with appreciable performance improvements. We refer the reader to section 2 for more details on the theory of layering as optimization decomposition.

Here, we argue that although layering as optimization decomposition formalism is rooted in the Network Utility Maximization problem, the complexity of issues involved in the web service selection problem closely resemble the NUM [32] problem. This argument sets the platform for us to extend *layering as optimization decomposition* to the web service selection problem. At a high level, in HMSCM, we map the composite service selection problem to NUM [32] problem as follows.

1. The service selection problem is a network partitioned into two main layers as Layer 1 and Layer 2.. Like in NUM [32], we put the service consumer at the forefront and have that the objective of Layer 2 is to minimize the financial burden and financial risk of a service consumer while accessing a service and to minimize the time it takes the consumer to access a composite service that meets their needs. Thus the utility function at Layer 2 should be the weighted sum over QoS attributes such as service execution cost, reputation, security and service response time etc.
2. We have that the objective of Layer 1 is to maximize the run time performance of the execution composite. Thus the utility function is the weighted sum of QoS values of QoS attributes such as service execution success, availability, response time, throughput etc.
3. Layer 2 provides "services" to layer 1. i.e the optimization solution from Layer 2 becomes the candidate solution space to Layer 1. The output of Layer 1 constitutes the best solution to the global problem. We refer to this approach as *"Top down service selection optimization"*.
4. In layered network design based on layering as optimization decomposition, the problem at each layer is formulated as some variant of the NUM [32] problem. In HMSCM, we develop "SLUM": "Service Layered Utility Maximization", a Layered version the MIP global planning model presented in [9]. Then each layer attempts to solve its local SLUM problem iterative over its local set of variables and problem inputs.
5. Thus each layer pursues to maximize its local utility and together the two layers strive contribute to global utility maximization.

Thus hence forth, we will interchangeably refer to Layer 1 as "Lower Layer" or "Service Provider Utility Maximization (SPUM) layer and to Layer 2 as "Upper Layer" or "Service Consumer Utility Maximization (SCUM) layer. It should be

noted that the local utility maximization in HMSCM fundamentally differs from the local planning approach described in section 1.3. In HMSCM locality is with respect to the entire "network" and not with respect to an individual task while in the latter locality is with respect to a task. It should be intuitive to see that in HMSCM it's still possible to express inter task constraints since optimization is being done by considering all "global" constraints within the scope of the layer. An elaborate discussion of the HMSCM model is in section 3.0

In order to evaluate our HMSCM model, experiments are currently being conducted on two different sets of WSDL web service composition problems that we have developed– one involving travel planning services and the other involving real time video streaming The performance efficiency and optimality are the metrics that will be used to compare HMSCM against the non-layered MIP solution.

1.7. Contributions

We fill the three gaps identified in section 1.4 by contributing the following knowledge to the state of the art:-

1. *Architectural, Design, Process and Practical Contributions*

We formulate a multilayer model for composite service selection based on the "theory" of *layering as optimization decomposition* as advanced by [28],[29],[30],[31]. Our architecture overcomes the three gaps identified in section 1.4 as follows:

- Like existing global planning techniques (but unlike existing local planning methods), in our approach, the service consumer is still able to express global task constraints within Layer 2. This closes the first gap.
- However, unlike in both current local planning and global planning approaches, the set of optimization variables within our architecture for which the user is required to specify weight preferences and constraints over is drastically reduced due to decomposition of optimization objectives i.e in current practice of QoS aware service composition, end users are required to specify weight preferences and constraints on variables such as reliability, throughput and so on. In our approach, service consumers can benefit from improvements in reliability, throughput optimization initiatives in Layer 1 without necessarily being aware of the process. Instead end users concern themselves only in QoS parameters that directly affect their financial burden or risk and speed of accessing a composite service. This fills the second gap
- Like in local planning (but unlike existing global planning strategies, this new model obtains the scalability benefit of the local planning models while filling the gap of the inability to express intertask constraints on one hand while overcoming the exponential state space explosion problem experienced in current global planning on the other hand. This is based on the observation that the problem complexity at each of the layers is drastically reduced when the initial set of optimization variables is decomposed. One may argue that the sequential nature of our layering approach introduces performance inefficiencies and hence, at least analytically, our approach may not be any better than non layered approaches. However, based on the theory that when decomposed subproblems are solved sequentially, improved performance results from the nonlinearity of problem complexity [21]. i.e a small change in the number of optimization variable leads to exponential change in the problem complexity. Thus the, the net effect of decomposition is larger than the inefficiencies introduced by sequential layering. This overcomes gap number 3 in section 1.4.
- Beyond filling the three gaps, we introduce the QoS "service provider view" of QoS aware service composition. This is motivated by the structure of the NUM [32] problem which has both the "network operator" optimization objectives and the "end user optimization objectives" at the core. We achieve the service provider view through the functions of Layer 1. This approach differs from all existing works, where QoS aware service selection is entirely viewed from the end user perspective, the consequence being that the user is overburdened in specifying weights and preferences over a large set of QoS some of which are too technical to make direct sense to an average user. The result of this separation of concerns is that at Layer 1, the engineers at the service provider can objectively and skillfully fix weight preferences over QoS attributes like reliability, throughput, and availability. The ultimate benefit to the service consumer is that they enjoy improved overall improved system performance efficiency resulting from layer 1 in a transparent manner. On the other hand, service providers can achieve their system performance objectives such as increased throughput, reliability etc.

We introduce two concepts: *Bottom up Problem Specification* and *Top down Service selection optimization*. The former refers to the process where the subproblems are defined starting at Layer 1 where service providers specify their optimization problem at design time by assigning weights to the various layer 1 QoS attributes and defining constraints on the optimization variables. This is followed by Layer 2 where at run time; end users specify their optimization objective by assigning weights on layer 2 QoS attributes and constraints on Layer 2 decision variables. In top down service selection optimization, we begin by solving for the utility maximization problem from Layer 2 (end user view point) followed by the utility maximization problem at

Layer 1 (service provider perspective). We explain our philosophy behind these two architectural decisions – bottom up problem specification and top down service selection optimization in section 3.

2. *Mathematical and Theoretical Contributions*

Multi-Objective formulation of the MIP problem for service selection. Informed by the structure of the model, we advance the original MIP model in [9] and related work that are single objective into two objective functions with the possibility of formulating any number of objective functions, each objective function addressing unique concerns in the service selection process. The result is an a mathematical model dub *"Service Layered Utility Maximization (SLUM)"*

1.8. Scope of the Study

HMSCM targets virtual enterprise brokers [1] operating within global virtual organizations dealing in ultra- low latency real time online services that are nowadays heavily driven by service oriented applications. Such virtual firms include but not limited to online travel planning, stock market and financial investment companies, virtual real time streaming multimedia content providers etc. The envisaged global virtual organization framework is that one in [1]. Although the proposed model is generic enough, in this paper, we assume workflow based service composition only as defined in Rao et al [33]. To simplify analysis without loss of generality, this paper will only focus on sequential workflows even though it should not be hard to extend it to other complex workflow patterns.

1.9. Outline of the Paper

The rest of this paper is organized as follows. In section 2 we review fundamental concepts in layering as optimization decomposition within the context of the NUM [32] problem. In so doing, reference is made to the OSI model .The goal is to extract common properties of the network design optimization problem that are extensible to the service composition problem. This allows us to make a meaningful formulation of the layering as optimization decomposition on service composition.

In section 3, we present the HMSCM model. First a qualitative description of the model is described emphasizing the mapping of the layering as optimization decomposition to our framework. Then, we go ahead to present the formal optimization model – the Service Layered Utility Maximization (SLUM) model. In section 4, closely related work is reviewed. We then finally make conclusions in section 5.

2. Layering as Optimization Decomposition

Given an original problem, decomposition entails restating the original problem into a set of independent or coordinated subproblems of smaller scale [34]. Because each subproblem is smaller than the original problem, decomposition yields more efficient solutions. The resultant subproblems can be solved either in parallel or in sequence. When the subproblems are sequentially solved, performance gain arises from that the observation that problem complexity is superlinear [21] i.e a small change in a factor such as the number of decision variables leads to disproportionate exponential change in the complexity of the problem.

There are many generic decomposition strategies and specific decomposition algorithms each suited for a well-known class of problems. There are hundreds of papers that comprehensively address the state of art in decomposition as an optimization method as well those that advance the state of the art. For instance, [34] reviews over 200 scholarly works on decomposition methods inclined to optimization problems in engineering but generally applicable to a wide range of scientific fields including computer science.

Layering as optimization decomposition [28],[29],[30],[31] is the latest formalism that has revolutionized the formulation of the Network Utility Maximization problem leading to improved network optimization strategies. The framework has following properties:

2.1. Properties of Layering as Optimization Decomposition [28-31]

i. Each network layer is viewed as a local subproblem whereas the network itself is the global optimization problem.

ii. Interfaces between layers represent either function of primal or dual variables. When two optimization problems are such that the there is a common variable, y in the objective functions of the two subproblems as in $f(x,y)$ *and* $f(x,z)$, then y is a primal or interface or complicating variable [21].The problem : *minimize* $f(x) = f1(u_1,y_1) + f2(u_2,y_2)$ *s.t* $y_1=y_2$ can be decomposed into two separate functions that are coupled by the constraint $y_1=y_2$. y_1, y_2 are the Lagrange dual constraints.

iii. The entire network is viewed as the "optimizer"

iv. A Network protocol at each of the layers is viewed as "a distributed solution to some global optimization problem". The global optimization problem is formulated in some form of the basic Network Utility Maximization problem.

v. Each of the layers iterate over a distinct subset of the global set of variables to achieve individual or local optimality. Overally, the individual protocols attempt to achieve a global objective.

vi. Each layer "serves" the layer above i.e

Table 1 below, shows the different layers (subproblems) and their corresponding objectives and solutions (protocols) in the TCP/IP network model.

Table 1. Multi-Layer Objectives in TCP/IP Network Design. Adapted from [31]

Layer	Optimization Objective	Solution
Application	Minimize response time	Various Application Protocols e.g HTTP,SFTP
Transport	Maximize Utility	TCP
Network	Minimize Path Cost	IP
Link	Reliability, Channel Access,	Various MAC protocols
Physical	Minimize Signal to Noise Ratio, Maximize Capacity etc	Various Physical Layer protocols

3. Hierarchical Multilayer Service Composition Model-HMSCM

3.1. Qualitative Description of the HMSCM Model

We cast the service selection problem onto the network utility maximization problem (NUM) based on the formalism of Layering as Optimization Decomposition as follows. First, we view the composite service composition as "a multi layered network" with each layer trying to achieve some local optimality towards to global optimization objective. In the case of network design, the global optimization problem is formulated as the basic NUM, generalized NUM or the stochastic NUM problem. Then based on the NUM global optimization problem, layered variants of the NUM [32] problem are formulated and solved. In the case of service composition, there is no universally agreed formulation of global "Service *Utility Maximization"* optimization model. However, as stated in section, the MIP global optimization model by Benatallah et al [9] has been widely adopted in the formulation of MIP solutions to service selection problems. In the place of NUM therefore we have what we dub here as "basic Service Utility Maximization (SUM)" model, referring to the MIP model in [9]. Then, we adapt the basic SUM model to fit the proposed layered architecture leading to SLUM for Service Layered Utility Maximization Model. SLUM is described in more detail in section 3.2.

Secondly, we have to identify the "layers" in our "network". Unfortunately, unlike in network design where there are well established network models such as OSI and TCP/IP, no network model or layered network formulation of the service selection problem exists today. Luckily we can draw some analogies from the NUM problem and based on existing work on QoS aware web service selection, we work backwards to identify a minimum number of "layers" in the network. Here goes the analogy. The generalized NUM problem puts the end user at the forefront leading to two types of functions [28]: 1) maximizing end users sum of utility functions over variables like rate, reliability, delay , jitter and 2) a network wide cost function determined by the network operator that can be functions of congestion, power efficiency etc. Putting the service consumer at the forefront, we can see at least two similar objective functions naturally arising in service composition problem. The following objectives can be identified: The first objective is that the service consumer would like to get access to the composite service at the minimum possible cost within the shortest possible time. Therefore from a consumer perspective minimization of *financial burden (which* includes minimizing actual cost of accessing the service and minimizing the financial risk) and *minimization of service response time are key concerns.* Financial risk is associated with QoS factors like reputation and security .Thus from this perspective, we have that the end user objective function is a utility function over the following web service QoS attributes: service execution cost, reputation, security and response time. On the other hand, the most important performance parameter from a business perspective is throughput –how many customers can be served in unit time. By implication, this extends to response time, reliability, availability etc. Therefore in the global virtual organization case, the virtual enterprise broker key objective is maximizing webservice total utility over throughput and other performance factors that affect throughput including response time, reliability, and availability. From these two objectives, we work backwards to formulate the two "layers" (subproblems) of our "network": Layer 1 and Layer 2. The objectives of these two layers were introduced in section 1 and here we summarize them in Table 2.

Table 2. Optimization Objective Functions in HMSCM

Layer	Optimization Objective	Solution
Layer 2	Maximize the utility function over composite webservice execution cost, reputation, security and response time	SCUM
Layer 1	Maximize the utility function over response time, service execution success, throughput ,availability	SPUM

Third, we need to establish which of the two layers "serves" the other. This is the same as asking the question: should the optimization process start at Layer 1 then Layer 2 (bottom up service selection optimization) or from layer 2 then layer 1 (top-down service selection optimization), does it matter which way? Starting with the last question, the answer is yes, the flow of information during the optimization process using the layered approach matters. Assume a top down approach is chosen. There is a possibility of selecting services with the lowest costs, lowest financial risk and lowest response time at Layer 2 but that have the worst reliability, reliability and or throughput when evaluated at Layer 1. Two possibilities: First if none of the composites meets the constraints at Layer 1, then no solution is found. Second, a subset or all the services may meet the threshold constraints on reliability and availability but only marginally. The result is that such services will have a higher probability of failure during execution whereas potentially more reliable but

more costly and less efficient services were "prematurely" dropped at layer 1. Conversely, if a bottom up optimization approach is followed, there is a possibility that composite services with the highest throughput, availability, reliability are chosen but may fail to meet the test at Layer 2 i.e either they do not meet cost, financial risk or response time constraints. If they did meet only marginally, the execution of the composite service may result in one of the following. Non responsiveness (web service takes too long to respond), a higher cost burdens to user, or potential loss of cash due to less trusted services. The point is that bottom up and top down optimization approaches may each yield different values to global optimization objective resulting into different optimality values. So whether to follow the bottom up or top down optimization approach is a problem itself. This paper takes a top down up approach –solve the SCUM problem first then afterwards the SPUM problem. The reason is that end users objectives remain at the core. i.e reducing financial burden, financial risk and reducing the time taken to access a service. The worry that less costly and more efficient but less reliable and low throughput services that are more likely to fail during execution can resolved by making the following observations. First, web service reliability is a function of availability among factors. A service that often fails during execution due to unavailability is less reliable. Fortunately, availability is a QoS factor that can be captured as part of the service level agreements (SLAs) between the virtual enterprise broker and the various virtual enterprises within the global virtual firm. The SLAs will ensure that variability in service availability across virtual enterprises is within acceptable bounds, if the virtual enterprises were to remain within the global virtual market. Secondly, services that are less responsive (large response times) have double impact. One is that potential timeouts definitely ruin the reliability of the service- failure to execute successfully. Second, the delay negatively impacts the overall system throughput. However, the bottom up optimization approach automatically mitigates these drawbacks – since the utility function accepts response time as part of the inputs and its output value is also restricted by the constraints on response time, it means that resultant services are not only of low financial burden but of high efficiency thus leading to overall increased throughput and reliability of the composition system. Even more, our proposed MIP optimization algorithm at Layer 2 attempts to find all feasible solutions that are then promoted to Layer 1. This is done so as to avoid early elimination of otherwise candidate web services with higher reliability, availability and throughput values. Thus the conclusion becomes that "Layer 2 serves Layer 1".

Fourth, we need to identify primal or Langrage dual variables between Layer 1 and Layer 2, if at all there are. We observe that response time is a "coupling or primal or interfacing variable" connecting Layer 1and Layer 2. We can eliminate the primal variable by maintaining this variable at only one of the two layers. Since optimization is done top down, we have that this variable is maintained at Layer 2 only. There are two main reasons for doing this. The first one is to maintain the validity of our choice of the bottom up optimization approach as explained in the preceding paragraph. The other and perhaps the most important reason is premised on the empirical evidence the response time for distributed software components exhibits time varying multimodal statistical distributions. We make allusions to [35]. By implication, the distribution of response time of web services is a stochastic process. Therefore, to increase chances of more efficient services being promoted from Layer 2 to Layer 1, response time must be one of the decision variables at Layer 2. In the end, Layer 1 and Layer 2 are decoupled in decision variables but coupled by data dependencies i.e Layer 1 has to wait for data from Layer 2.

Fifth, we model the flow of data or information from one network layer to the next layer as the flow of the web service composition Bipertite graph. The original graph contains all candidate web services. As the graph flows through Layer 2 and Layer 1, some services are eliminated.

3.2. Mathematical Formulation of the HMSCM Model

3.2.1. Problem Formalization

We formally restate the composite service selection problem as follows:

Given the tuple, $\langle R, F, G \rangle$

Find: $P^b \in G$ that can execute F *to* satisfy *R*

Where;

- *R* is the complex service request such that $R = \langle r_1, r_2, \ldots, r_n \rangle$ where r_k is an atomic service request within *R*.
- *F* is a sequential abstract workflow such that $F = \langle t_1, t_2 .., t_n \rangle$ where t_k is a workflow task within *F* such that the execution of t_k leads to the fulfillment of r_k. The tasks are sequentially ordered as t1 $\rightarrow t_2 \rightarrow_{,..,} \rightarrow t_n$,
- *G* is the web service composition Bipertite graph such that *G* is the N-tuple $\langle V_1, V_2, \ldots, V_n \rangle$ where V_k is a vertex set containing a list of functionally similar concrete web services that can execute the task t_k . Therefore V_k is the data structure $List<W_{kj}>$ where W_{kj} is the j^{th} service in $V_{k..}$ Each W_{kj} can be defined by the tuple $\langle I, O, Q \rangle$ where *I* is the set of input parameters, O is the set of output parameters and Q is the set of QoS values associated with W_{kj}. Any complete path constituted by a service drawn from V_1, and another service from V_2 , ..., and finally another service from V_n constitutes a candidate solution. If *m* is the number of services in every $V_{k..}$ then as shown earlier in section 1, there exists m^n such candidate solutions or candidate composite services. Therefore we need to find P^b, the best path (composite service) that satisfies *R*.

This paper is about solving for P^b. In section 3.2.2 we elaborate how our HMSCM model finds P^b using the Service

Layered Utility Maximization (SLUM) algorithm.

3.2.2. The Service Layered Utility Maximization Solution

Table 3. The set Q of Web service QoS Attributes

QoS Name	Layer	Absolute Symbol	Atomic Service Symbol	Composite Service Symbol
Reliability	1	r	r^s	r^c
Availability	1	a	a^s	a^c
Throughput	1	h	h^s	h^c
Execution Duration	2	d	d^s	d^c
Execution Cost	2	c	c^s	c^c
Reputation	2	u	u^s	u^c
Security	2	z	z^s	z^c

Table 4. Composite Service QoS Aggregation Functions based on Sequential Workflows

QoS Name	Aggregation Function
Reliability	$r^c = \prod_{I=1}^{i=N} r^s$
Availability	$a^c = \prod_{I=1}^{i=N} a^s$
Throughput	$h^c = 1/N(\sum_{I=1}^{i=N} h^s)$
Execution Duration	$d^c = \sum_{I=1}^{i=N} d^s$
Execution Cost	$c^c = \sum_{I=1}^{i=N} c^s$
Reputation	$u^c = 1/N(\sum_{I=N}^{i=N} u^s)$
Security	$z^c = 1/N(\sum_{i=1}^{i=N} z^s$

Table 3 above contains some of the relevant web service QoS attributes as defined in various literature sources such as [6-9]. We have also assigned operational absolute symbol, symbol of QoS parameter when referring to an atomic service as well as when referring to a composite service. Hence forth, we will use these QoS variables to define our model although the methods presented based on this sample of QoS parameters are general enough to be applicable beyond the sample parameters used in this paper. In Table 4, we provide aggregation functions for computing the overall QoS of a composite service or a plan assuming sequential workflows.

SLUM contains three major phases: QoS vector decomposition phase, Bottom Up Problem Specification phase and the Top Down Service Selection phase. Bottom up problem specification entails defining preferences on QoS parameters, formulating an optimization objective function and defining optimization constraints starting with Layer 1 then Layer 2. The Top down service selection phase involves solving the Service Consumer Utility Maximization (SCUM) subproblem at Layer 2 followed by a solution to the Service Provider Utility Maximization (SPUM) subproblem at Layer 1.

1. Decomposition of the Set of Quality Attributes

The original set of web service QoS attributes, Q is divided initially into two disjoint partially layered sets of QoS attributes, Q1 and Q_2, such that Q_1 is in Layer 1 and Q_2 is assigned to Layer 2. Q2 contains all web service QoS parameters related to the financial burden and financial risk to be borne by the service consumer and one performance QoS parameter – response time. Q_1 contains the set of all performance parameters except response time. In practical SOA applications, this step should be performed by the Virtual Enterprise Broker.

2. Bottom Up Problem Specification

I. SPUM Problem Specification at Layer 1

A. Layer 1 Weight Assignment to QoS Parameters

As a first step, the Virtual Enterprise Broker should define a weight vector W_1 in which the *i*th element corresponds to a weight assigned to the *ith* QoS element in Q_1 such that $\sum_{j=1}^{j=n} W_1^j = 1$. A weight value assigned to a Qos parameter in Q_1 indicates the relative priority of that QoS attribute from a service providers point of view. Suppose W_1 = [0.5, 0.2, and 0.3] for reliability, availability and throughput respectively, then it means that the service provider is concerned about service reliability more than any other QoS attribute. From the same example, the virtual enterprise broker prefers services with a higher throughput than service which may have a higher availability with smaller throughput values. The weights can be adjusted as service performance statistics evolve over time.

B. Layer 1 Objective Function Definition

At layer 1, the objective function of the SUM problem, F_1 is to maximize the utility function U_1 over the set Q1, given the initial web service graph G, the weight vector W_1, the set decision variables X_1 subject to a set of constraints C^1. X_1 contains the set of decision variables at Layer 1, while C is the set of constraints on X_1. The objective is captured according (1) and refined according to (2).

$$F_1 = \text{maximize}[\![U_1(M_1^c, W_1)]\!] \quad (1)$$

The objective function F_1 in (1) is translated as : maximize the value of the utility function U_1 which takes as input, the QoS matrix M_1^c and the weight vector W_1. M_1^c is the matrix containing normalized aggregate QoS values for each candidate composite service (plan) on every QoS attribute in Q1. i.e by adopting a notation similar the one used in [9], the rows represent a candidate execution plan and the columns represent the jth QoS attribute and M_1^{cij} is the raw aggregate j^{th} QoS value of the i^{th} execution plan . To compute M_k^{cij}, the aggregation functions given in Table 4 are used accordingly.

Note that some QoS parameters can be positive while others negative. The QoS of positive parameters increase with increasing values of the parameter. The QoS of a negative parameter decline with increasing value of the attribute. For example in Table 3 above, execution duration and execution cost are both negative QoS attributes and the rest are positive parameters. For this reason, the matrix M_1^c needs to be normalized. If M_k^{cij} is a positive parameter, we denote the normalized image of M_k^{cij} by M_k^{cij+} or

M_k^{cij-} otherwise. M_k^{cij+} and M_k^{cij-} are computed according to the scaling functions given in (2) and (3) respectively.

$$M_k^{cij+} = [\![M_k^{cij} - M_k^{cjmin}]\!]/[\![M_k^{cjmax} - M_k^{cjmin}]\!] \quad (2)$$

$$M_k^{cij-} = [\![M_k^{cjmax} - M_k^{cij}]\!]/[\![M_k^{cjmax} - M_k^{cjmin}]\!] \quad (3)$$

In both (2) and (3) :

- If $M_k^{cjmax} - M_k^{cjmax} = 0$,1 is returned.
- M_k^{cjmax} is the maximum value in the jth column
- M_k^{cjmin} is the minimum value in the jth column
- k, as usual is the optimization layer 1 or layer 2

We will denote the resultant matrix after scaling the matrix M_k^c by $M_k^{c\prime}$.Thus the optimization objective function at layer 1 is revised to (4).

$$F_1 = \text{maximize}[\![U_1(M_1^{c\prime}, W_1)\ ]\!] \quad (4)$$

By applying the Simple Additive Weighting, SAW [18] to (4) as our utility function, (5) holds.

$$F_1 = maximize[\![M_1^{c\prime} * W_1]\!] \quad (5)$$

Equation (5) can be expanded to (6). (6) holds because in our case all layer 1 QoS variables are positive.

$$F_1 = maximize[\![\sum_{j=1}^{j=3}[\ M_1^{cij+} * W_1^j]]\!] \quad (6)$$

A. Definition of Layer 1 Optimization Constraints

Let *R, A* and *H* be the reliability, availability and throughput thresholds set by the virtual enterprise broker on every execution plan. We use the notation C^{ki} to denote the i^{th} constraint at the k^{th} layer. When *k=1*, the following constraints are enforced. We have:

$$C^{11}: \quad r^c \geq R \quad \text{or} \quad \prod_{i=1}^{i=N} r^s \quad \geq R \quad (7)$$

Since C^{11}: is nonlinear, we linearize it by taking the logarithms on both the L.H.S and R.H.S of (7) to get (8).

$$C^{11}: \log r^c = \sum_{i=1}^{i=N} \log(r^s) \geq log\ R \quad (8)$$

C^{11}, as represented in (8) is the constraint on composite service reliability.

Similar to C^{11}, C^{12}, the availability constraint on composite service availability is expressed according to (9).

$$C^{12}: \log a^c = \sum_{i=1}^{i=N} \log(a^s) \geq log\ A \quad (9)$$

The constraint on composite service throughput at the SPUM layer is captured in (10).

$$C^{13}: h^c = 1/N(\sum_{i=1}^{i=N} h^s) \geq H \quad (10)$$

We need a binary variable to indicate whether or not a web service WS_{ji} is selected from the vertex set $V_i \in G$ to execute a workflow task, t_i. Conventionally this variable is represented as y_{ij} . In this work, we will represent this variable as y_k^{ij} to reflect our layered architecture, where *k* is the layer number. At *k=1*, constraints C^{14} and C^{15} hold. C^{14} indicates that a service can assume y_1^{ij} value of 1 or a y_1^{ij} value of zero. In (12), C^{15} dictates that only one service can be selected from each vertex set V_i to execute a task t_i in the set F of workflow tasks.

$$C^{14}: 0 \leq y_1^{ij} \leq 1 \quad (11)$$

$$C^{15}: \sum y_1^{ij} = 1, i \in V_i, \forall i \in \text{F} \quad (12)$$

In addition to the above constraints, at layer 2, we introduce the binary variable l_2^{ij}. l_2^{ij} indicates whether or not the service WS^{ij} was selected during layer 2 SCUM optimization process. We enforce the constraint in (13) to imply that only services previously selected during layer 2 optimization should be selected.

$$C^{16}: 0 \leq e_2^{ij} = 1 \quad (13)$$

Thus the set of optimization constraints C^1 at layer 1 contains C^{11}, C^{12}, C^{13}, C^{14}, C^{15}, C^{16}:

II. SCUM Problem Specification at Layer 2

A. Layer 2 Weight Assignment to QoS Parameters

As a first step, the service consumer should define a weight vector W_2 in which the *i*th element corresponds to a weight assigned to the *ith* QoS element in Q_2 such that $\sum_{j=1}^{j=n} W_2^j = 1$. A weight value assigned to a Qos parameter in Q_2 indicates the relative priority of that QoS attribute from a service consumer point of view. Suppose W_2 = [0.1, 0.4, 0.3, and 0.2] for execution duration, execution Cost, reputation and security respectively, then it means that the service consumer cares about cost more than any other QoS attribute. Recall that this differs from the state of the art where the end user is always assumed to be responsible for specifying weight preferences over all QoS attributes. With our approach, the end user can benefit from the optimization of parameters such as throughput, reliability and availability without necessarily being aware of the optimization process surrounding these parameters, just in the same way in the NUM problem, the end user can benefit from improved physical layer forward error correcting codes while such details are abstracted from them. After all, all service consumers always expect that whenever they access a service it's available and that it will execute successfully all the time. Consequently with our methodology, end users have fewer QoS attributes over which to specify weights.

B. Layer 2 Objective Function Definition

At layer 2, the objective function of the SCUM problem, F_2 is to maximize the utility function U_2 over the set Q2, given the web service graph G^1, the set decision variables X_2 subject to a set of constraints C^2. X_2 contains the set of decision variables at Layer 2 and C_2 is the set of constraints on X_2. $G^1 \sqsubseteq G$ i.e G^1 is the set of feasible solutions from Layer 1 or the set of candidate solutions at Layer 2. G^1 may contain all or just a subset of paths from the original graph, *G*. This objective function is stated according to (14).

$$F_2 = maximize[\![U_2(M_2^c, W_2)\ ]\!] \quad (14)$$

By applying (2) and (3) and using the conventions adopted in this paper, (14) transforms to (15). The objective function in (15) holds since at layer 2 duration and cost are negative parameters while reputation and security are positive

parameters.

$$F_2 = maximize[\sum_{j=1}^{j=2}[\ M_2^{cij-} * W_2^j] + \sum_{j=3}^{j=4}[\ M_2^{cij+} * W_2^j]\ ] \quad (15)$$

C . Layer 2 Definition of Optimization Constraints

Let *D, C, U* and *Z* be the extreme values set by the service consumer on composite service execution response time, execution cost, reputation and security in that order. Here we define the constraints on composite service execution duration, execution cost, reputation and security in (16), (17), (18) and (19) respectively.

$$C^{21}: d^c = \left(\sum_{i=1}^{i=N}\ d^s\right) \le D \quad (16)$$

$$C^{22}: c^c = \left(\sum_{i=1}^{i=N}\ c^s\right) \le C \quad (17)$$

$$C^{23}: u^c = \left(\sum_{i=1}^{i=N}\ u^s\right) \ge U \quad (18)$$

$$C^{24}: z^c = \left(\sum_{i=1}^{i=N}\ z^s\right) \ge Z \quad (19)$$

In (16), (17),(18) and (19) the service consumer expects the best composite service:-

- Not to take more than D seconds before the consumer gets the final results to their service request as conveyed by C^{21}
- To cost them not more than *C* units of money to access the business service provided by the technical composite service as captured by C^{22}
- To have an average reputation of at least U on the interval [1, 5].
- To have a security rating of not less *Z* on the average. The security associated with accessing the business service in this case is the average of the each service provided by each virtual enterprise.

Just like with Layer 1, constraint on the allocation constraint y_{ij} are defined. Adopting our notation, we have (20) and (21) with the usual meanings.

$$C^{16}: 0 \le y_2^{ij} \le 1 \quad (20)$$

$$C^{17}: \sum y_2^{ij} = 1, i \in V_i\ , \forall i \in \mathrm{F} \quad (21)$$

Top Down Service Selection

A. SCUM Optimization Process at Layer 2

At layer 2, all feasible solutions are determined i.e all combination of services that can fulfill the objective function F_1 subject to the constraints set C^1 are returned in a solution pool. The reason for obtaining all feasible solutions as opposed to the optimal solution is so as to prevent possibility of prematurely dropping a service which would have otherwise scored better than a majority of the selected services.

We define a Web service to Task Assignment Matrix (STAM. At layer 2, we will denote this matrix by L_1.As an example, consider a two task workflow. Suppose initially before selection there were 3 candidate services per task.

Before layer 1 evaluation, this matrix is represented in tabular form as in table 5 and after Layer 1 Optimization the matrix L_1 is represented as shown in table 6 below.

Table 5. An example Web service to Task Assignment Matrix for *m*=3, *n=2* before Layer 1 Optimization

Workflow Task, *i*	Candidate Web service, *j*		
	1	2	3
1	0	0	0
	0	0	0

Table 6. An example Web service to Task Assignment Matrix for *m*=3, *n=2*, after Layer 1 Optimization

Workflow Task, *i*	Web service, *j*		
	1	2	3
1	1	1	0
2	1	0	1

During optimization at layer 1, for each service s_{ij} that is selected and assigned to a task *i* , *yij* is updated to *1*.Suppose the resultant web service to task assignment matrix after layer 2 optimization is as shown in table 6. The Web Service to Task Assignment matrix, L1 in table 6 indicates that:-

- That services $S_{11,}$ S_{12} were selected for task 1 while service S13 was not selected for task 1 after phase 1 optimization.
- Service S21 and S23 were selected for task 2 while service S22 was left out
- Out of 9 candidate solutions, only 4 feasible solutions were found. In this case only the paths $< S_{11,}, S_{21} >$, $< S_{11,} S_{23} >$, $<S_{12,} S_{21} >$ and $<S_{12,} S_{23} >$ will be evaluated for performance at layer 2.

Thus during SPUM optimization process at Layer 1, the e_2^{ij} values of $S_{11,}$, $S_{12,}$ $S_{21,}$ S_{23} will 1 and only these services will be evaluated at Layer 2.

B. Layer 1 -SPUM Selection Optimization Process

Having selected services whose combination maximizes the utility of user preferences on service execution cost, reputation etc and that meet the constraints defined on cost, reputation , at layer 1 the goal is to select the service combination that maximizes utility on performance related QoS subject to constraints defined on the performance QoS variables. The output of layer 1 optimization process is therefore a set of service combinations that fulfill requirements of both layer 1 and layer 2. The solution at Layer 2 therefore constitutes P^b.

4. Related Work

As stated earlier the selection of the best composite service from a large pool of services based on many QoS attributes is a Multi-Criteria Decision Making NP hard problem. Thus many researchers are attempting to attack the problem from a MCDM perspective using different techniques. We divide prior work into two categories – the

first one consisting of approaches that do not use any decomposition technique here in called *monolithic Multi-Criteria Optimization Models* and strategies which employ some decomposition strategy herein referred to *Decomposed Multi-Criteria Optimization models.*

4.1. Monolithic Multi-Criteria Optimization Models

In [40] QoS based service composition method based on constraint programming using simple additive weights and mixed integer programming is presented. Like the MIP model presented in [9], this method can be shown to be more efficient than exhaustive search planning approaches. However, still, it remains unscalable for large scale service composition and where many QoS attributes are considered.

A planning graph based approach based on multiple criteria is proposed in [42]. Here the composition problem is specified using the PDDL language. OWL-S plan is then used to generate all possible execution path where each path is possible solution (composite service) satisfying the service composition goal. Then the SAW [18] is used to compute the overall score of each plan and the best plan generated. As pointed out in section 1, optimization based on propositional logic is limited; not all problems for instance can be modelled in PDDL efficiently. More, the algorithm presented here uses exhaustive search to generate all plans making exponential time in nature.

In [39] a multiple criteria method for service selection based on Fuzzy logic is presented. Here users express constraints as Fuzzy rules. A weighting approach is also used where a user assigns what the authors call a Confidence Factor (CF) on the range [0, 1] where the CF denotes the importance of each fuzzy rule. Thus rules that are more important from a user's point of view are assigned a higher CF value than the less important ones. In real life applications, having users encode their preferences on QoS attributes in form of Fuzzy rules is far from practical. Also this method severely suffers combinatorial sate space explosion when the number of QoS attributes increases, the rule base grows exponentially.

Virginie G. et al in [41] provide a linear programming method for QoS web service composition. However, the method is still based on monolithic optimization models that are not scalable with increasing number of web service QoS attributes as well as the size of web services.

A global optimization method using Taylor expansion based on MIP is presented by Fan Yan [43].However; neither do the authors provide a justification of the use of the Taylor expansion nor results to back up their model. More as explained earlier, the scalability of MIP models are limited to small scale service composition [20], [22.]

Shiang Chia Liu in [36] proposes a genetic algorithm for composite service selection. The advantage this has over MIP based models is that Genetic Algorithms (GA) are more efficient for ultra large problem sizes. However, GAs require configuration and tuning of extra parameters such as the population size [44].

The main innovation by Ngoko Y et al [44] is a MIP global optimization model for workflow based service compositions involving multiple cooperating abstract composite service services. Moreover, the optimization model takes into account service level agreement constraints. Generally the authors empirically show that their MIP global planning model is more desirable than local planning in terms of optimality of solutions. However, only two QoS attributes are considered during optimization so it remains unknown how the model can behave as the number of QoS factor s grow larger.

All the foregoing algorithms suffer from the following. They are single objective and all assume that all weighting and preferences over QoS attributes is end user guided. This is too burdensome to the user. Secondly, the methods have a single objective or utility function iterating over the entire global set of variables. As we saw earlier, this leads to combinatorial explosion.

4.2. Decomposed Multi-Criteria Optimization Models

Singh et al [11] provides a decomposition method for service selection based on mixed integer programming. User global constraints are transformed into local constraints. Although the authors claim reduced MIP model that can be solved in linear time, no details are provided on how the decomposition method works.

A phased approach to MIP optimization closer to ours is presented in Alrifai Mohamad [37]. The method is based on decomposition of global constraints into local constraints that then used during a local optimization phase. In phase 1 one each QoS attribute value into quantity levels for each service in each service class. Then the objective is to find the best combination values that will be used as upper bound constraints within the second phase that employs a local planning approach. MIP is applied to find the best combination of values that satisfy the constraint. In phase2, local search is used to select the best services. The challenge with this approach is that expressing global constraints that will not be violated by local constraints is a challenge. Further, the performance of the model is affected by the number of quantity levels d . The larger the d the less efficient the model becomes and vice versa. The value of d or range of d for which the model can perform better than conventional MIP remains unknown.

We take a different approach to decomposition of the MIP approach to optimizing service selection. We base our work on a well-founded theory "Layering as Optimization Decomposition". Here we are not only decomposing constraints but decomposing the network level objective into layerwise local objective functions that address both the concerns of the service provider and the service consumer. Like in [37], our approach is a two-step process. However unlike in [37] and others, our optimization in step (layer in our case) is based on global planning method in [9] (no task level local planning service selection is done). Thus we use global planning at each layer guaranteeing no violation of

global constraints, while enjoying improved performance due to smaller set of optimization variables at each layer.

To the best of our knowledge, this first model: 1) that jointly considers service consumer utility maximization and service provider utility maximization under a single framework. 2) that extends formal theory of Layering as optimization decomposition [28-31] to the joint service provider and service consumer utility maximization in the web service selection problem.

5. Conclusion and Ongoing Work

Service composition continues to be acknowledged as the most agile technology approach to support dynamic business to business collaborations such as those found in global virtual organizations. The high demand on VEs to respond reliably and in time to consumer requests cannot be overemphasized. Yet, composing services efficiently in dynamic environments such as global Virtual Organizations still remains a formidable challenge. Current approaches besides being too sophisticated for industrial adoption often introduce performance overheads due to combinatorial explosion and severely fall short in large scale composition contexts. We have contributed to body of knowledge by proposing three key approaches to services composition considering end user objectives and service provider objectives 1) A Generic Layered Incremental Model that partitions the composition problem into 2 layers based on the theory of layering as optimization decomposition – one layer maximizing local utility of the service consumer while the other maximizing the local utility of the service provider. Together both layers attempt to achieve a global objective which is efficient service composition meeting user needs, 2) We develop a MIP optimization model called the Service Layered Utility Maximization (SLUM) extending the MIP model in [9] and formulate the problem at each layer in form of SLUM. Particularly, we introduce two submodels of SLUM – Service Consumer Utility Maximization (SCUM) and Service Provider Utility Maximization (SUM) addressing consumer and provider needs respectively. By dividing the problem into separate but interdependent layers, the combinatorial space explosion problem is attacked by reduced the number of variables to be combined. Further the model still supports global constraints. 3).Motivated by the NUM [32], the portioning of QoS variables into service provider facing and user facing, relieves the end user of the unnecessary burden of weighting QoS factors while still giving them to opportunity to maximize their own utilities on QoS attributes that matter to them. On the other hand, using our model, service providers such as the virtual enterprise brokers, have the opportunity to objectively influence the service selection process by controlling low level performance factors such as throughput, reliability and availability. The benefits of such an objective tuning of the service selection is passed over transparently to the end users.

Currently we are working on validating the visibility of our models quantitatively using prototypes we have developed internally. We target to publish our preliminary results within the next two months. We hope that these results will shape the future direction on the applicability of the theory of layering as optimization decomposition to what we have coined the "Service Utility Maximization (SUM)" problem within the web services research arena.

REFERENCES

[1] Molina A. and Flores M., "A Virtual Enterprise in Mexico: From Concepts to Practice", Journal of Intelligent and Robotics Systems, 26: 289-302, 1999

[2] Amit G., Heinz S .and David G. (2010). Formal Models of Virtual Enterprise Architecture: Motivations and Approaches, PACIS 2010 Proceedings.

[3] Cammarimha M. and Arfsamanesh H.(2007). A comprehensive modelling framework for collaborative networked organizations, Journal of Intelligent Manufacturing, Springer Publisher,Vol.18(5),pp-527-615

[4] Arfsamanesh H. et al (2012). A framework for automated service composition in collaborative networks, FNWII: Informatics Institute, http://hdl.handle.net/11245/1.377/099

[5] Dustdar S. and Wolfgang S. (2005). A Survey on Web Services Composition

[6] Kuyoro Shade O. et al (2012). Quality of Service (QoS) Issues in Webservices, International Journal of Computer Science and Network Security, Vol 12 (1), January, 2012.

[7] Mahboobeh M. and Joseph G.D (2011). Service Selection in Webservice Composition. A comparative Review of Existing Approaches, Springer- Verlag Berlin, Heidelberg, 2011

[8] Rajendran T. and Balasubramanie P. (2009). Analysis on the Study of QoS Aware Webservices Discovery, Journal of Computing Vol. 1(2), December, 2009.

[9] Benatallah B. et al (2005). QoS-Aware Middleware for Web Services Composition. IEE Transactions on Software Engineering, Vol. 30, No. 5, May 2005.

[10] Bin Xu et al (2011). Towards Efficiency of QoS driven semantic webservice composition for large scale service oriented systems, Sringer, 211, DOI 10.1007/s11761-011-0085-8

[11] Singh K.A (2012). Global Optimization and Integer Programming Networks. International Journal of Information and Communication Technology Research

[12] Peter Bartalos, M´aria Bielikova (2011). Automatic Dynamic Web Service Composition: A Survey and Problem Formalization, Computing and Informatics Journal, Vol. 30, 2011, 793–827

[13] Jacob Nielsen (1993). Usability Engineering, Morgan Kaufmann, 1st Edition, September, 1993.

[14] September 14, 2009 - Akamai Reveals 2 Seconds as the New Threshold of Acceptability for eCommerce Web Page

Response Times
http://www.akamai.com/html/about/press/releases/2009/press_091409.html, Last Accessed 4th April,2015

[15] Alberto Savoia (2009). Webpage Response Time. Understanding and Measuring Performance Test Results. http://ericgoldsmith.com/wp-content/uploads/2009/02/web_page_response_time_101.pdf.

[16] http://www.nngroup.com/articles/response-times-3-important-limits/, updated 2014, Last accessed on 4th April,2015.

[17] Nah 4. (2004). A Study on tolerable waiting time. How long are web users willing to wait ? Behaviour and Information Technology , Forthcoming

[18] HC-L and K.Yoon (1981). Multiple Criteria Decision Making, Lecture Notes in Economics and Mathematical Systems, Springer Verlag. J Op. Res Soc. Vol 49(3), pp 237-252, March 1998.

[19] Toni Mancini, Piere Flener, Amir Hossein Monshi and Justin Pearson (2009). Constraint Optimization over Massive Databases in Proceedings of the 16th International Conference RCRA workshop (RCRA 2009).

[20] Seog Chan Oh, Dongwon Lee, and Soundar R. T. Kumara (2006). A comparative illustration of AI planning-based web services composition, ACM

[21] Stephen Byod, Lin Xiao and Almir Mutapcic (2003). Notes on Decomposition Methods, Notes for EE3920, Stanford University, Autum available at 2003 http://web.stanford.edu/class/ee392o/decomposition.pdf, Last accessed 4th April, 2015.

[22] Matthew Kitching (2010) .Decomposition and Symmetry in Constraint Optimization Problems, PhD Thesis, Graduate Department of Computer Science, University of Toronto.

[23] Stephen Cook (1971). The Complexity of Theorem Proving Procedures, STOC 71, Proceeding of the third ACM Symposium on the Theory of Computing, pp 151-158,1971

[24] Kautz H. and Selman B. (1992). Planning as Satisfiability, available online at http://www.cs.cornell.edu/selman/papers/pdf/92.ecai.satplan.pdf, Last accessed on 5th April, 2015

[25] Kautz H. and Selman B.(2004). WALKSAT in the 2004 SAT Competition, available online at http://www.cs.rochester.edu/~kautz/papers/walksat.pdf,

[26] Blum L. Avrim and Furst L. Merrick (1997). Fast Planning Through Planning Graph Analysis, Artificial Intelligence, 90:281-300, 1997

[27] Albert Rainer and J. Urgen Dorn (2009). MOVE: a generic service composition framework for Service Oriented Architectures. IEE Webservices Challenge 2009.

[28] Chiang Mung. (2006). Layering as Optimization Decomposition, Electrical Engineering Department, Princeton University. Also available online at http://www.ece.rice.edu/ctw2006/talks/ctw06-chiang.pdf

[29] Chiang M. et al. (n.d). Layering as Optimization Decomposition. Current Status and Open Issues, Electrical Engineering Department, Princeton University.

[30] Chiang M. et al. Layering as Optimization Decomposition. Ten Questions and Answers, available at http://web.stanford.edu/class/ee360/previous/suppRead/read1/layer_1.pdf

[31] Steve Low (2013). Scalable Distributed Control of Networks of DER, Computing & Math Sciences and Electrical Engineering, Caltech University

[32] Kelly F.P, Maulloh A and Tan T(1998). Rate Control for Communication Networks. Shadow Prices, Proportional Fairness and Stability.

[33] Rao Jinghai and Xiaomeng Su (2004). A Survey of Automated Web Service Composition Methods.

[34] Shan S. and Gang G.G (2009). Survey of Modeling and Optimization Strategies for High Dimensional Design Problems with Computationally Expensive Black Box Functions, Springer Verlag, Published Online August, 2009.

[35] Kounev S., Gorton I and Sachs K. (Eds).SIPEW 2008, LNCS, 5119,283-302, 2008, Springer Verlag.

[36] Shiang Chia Liu (2012). Applying Genetic Algorithm to Select Webservices Based on Workflow Quality of Service, Journal of Electronic Commerce,Vol 13(2), 2012

[37] Alrifai Mohammad (n.d) Distributed and Scalable QoS Optimization for Webservices Composition, PhD Thesis, L3S Research Center, Leibniz University of Hannover, Germany

[38] Amel Boustil, Nicholas Sabouret and Ramdane Maamri (2010). Webservices Composition Handling User Constraints. A semantic approach.

[39] Mahdi Bakhshi and Seyyed Mohsen Hashemi (2012). User Centric Optimization for Constraint Webservice Composition Using a Fuzzy Guided Genetic Algorithm System. International Journal on Webservice Computing Vol. 3, No 3., September 2012.

[40] Mohammad Alifarai, Dimitrios Skoutas and Thomas Risse (2010). Selecting Skyline Services for QoS based Webservice Composition, April 26-30, Raleigh NC, USA.

[41] Virginie G. et al (2013). A linear Program for QoS webservice composition based on complex workflow. 2013.

[42] Shanliang Pan and Quinjiao Mao (2013). Case Study on Webservices Composition Based on Multi-Agent System. Journal of Software, Vol. 8, No 4. April 2013.

[43] Fan Yan (2012). Global Optimization Method for Webservices composition based on QoS, International Conference on Engineering and Business Management, 2012.

[44] Ngoko Y., Goldman A, and Milojicic D. (2013). Service Selection in Webservice Compositions Optimizing Energy Consumption and Service Response.