Scalable Computing: Practice and Experience
Volume 13, Number 4, pp. 315–324. http://www.scpe.org
ISSN 1895-1767

A SIMULATION PLATFORM FOR EVALUATION AND OPTIMIZATION OF COMPOSITE APPLICATIONS

JĀNIS GRABIS* AND MARTINS BONDERS

Abstract. Composite applications are developed by integrating independent web services and deployed in a dynamic cloud based environment. An ability to modify the composite applications in response to changing business needs significantly contributes to agility of enterprise information systems. Deployment and execution in the cloud based environment allows to requisition resources necessary for efficient execution of the composite applications. However, properties of the composite applications directly depend upon characteristics of external services used and environmental factors, which in the case of public networks, exhibit high degree of variability. In order to address this issue, the objective of this paper is to develop a simulation and business process modelling based platform for evaluation of composite applications to ensure that the applications developed deliver expected performance. The combined approach allows for comprehensive evaluation subject to stochastic and dynamic factors, and the platform integration reduces the modelling overhead. Application of the simulation platform is demonstrated using an example of designing a composite application for a taxi call center.

Key words: Composite applications, optimization, simulation, web service selection

1. Introduction.

Composite applications are developed by combining existing information technology resources to provide new business capabilities [15]. They are characterized by a high level of flexibility and agility and can be set up relatively quickly to capture new business opportunities or to adjust to changes in business processes. Most frequently composite applications are designed by composing external services such as web services. This fact allows to attain benefits associate with software assets reuse and to reduce infrastructure maintenance efforts. However, properties of the composite applications directly depend upon characteristics of external services used and environmental factors, which in the case of public networks, exhibit high degree of variability. Therefore, the selection of appropriate and reliable services is of major importance. Multiple methods have been elaborate for selection of such services from the set of candidate services providing similar functionality [17]. These methods often use Quality-of-Service (QoS) measurements as selection criteria and rely on optimization techniques for choosing the appropriate web services. This approach has a number of limitations. Optimization techniques have well-known limitations [14] and they cannot account for all factors affecting the service selection, particularly, stochastic and dynamic factors. The selection process is also decoupled from the design process of composite applications. Therefore, the web services selection might not adequately represent performance of the composite application as a whole and there could be a significant overhead associated with the web service selection process leading to increased effort and reduced agility of developed composite applications.

In order to address these limitations, the objective of this paper is to elaborate a platform for comprehensive evaluation of composite applications. The evaluation should ensure that applications developed deliver the expected performance.The platform combines optimization techniques with simulation for the selection of services and the evaluation of the composite application. It also uses a business process model underlying the composite application to be developed as the evaluation basis to reduce effort associated with development of multiple evaluation models. The platform includes a module for web service selection and a module for simulation of performance of the composite application depending upon the web services selected and environmental parameters. The web services are selected using the mathematical programming model, which accounts for both functional and non-functional requirements expressed in the terms of costs associate with application usage. The simulation module evaluates expected performance of the composite application and identifies key requirements for the execution requirement. The main contributions of this paper to the state of art are: 1) accounting for both functional and non-functional factors in the service selection; 2) providing of the simulation environment for evaluation of performance of the composite applications; and 3) integration of the simulation and optimization models with business process and executable process models. Application of the simulation platform is demonstrated using an example of designing a composite application for taxi call center.

The rest of paper is organized as follows. Section 2 reviews related research. The evaluation platform is introduced in Section 3. Section 4 elaborates optimization and simulation models used for evaluation of composite applications. Section 5 demonstrates application of the platform, and Section 6 concludes.

*Institute of Information Technology, Riga Technical University, Kalku 1, Riga, LV-1658, Latvia, (grabis@iti.rtu.lv).

Table 2.1: Overview of Web service selection methods.

Source	Method
Canfora et al. (2008) [5]	Genetic Algorithms
Cai et al. (2009) [4]	Artificial Neural Network
Hou and Su (2006) [8]	Analytic hierarchy process (AHP)
Huang et al. (2009) [9]	Linear programming techniques for Multidimensional Analysis of Preference
Lin et al. (2008) [11]	QoS Consensus Moderation Approach
Ma and Zhang (2008) [12]	Convergent population diversity handling genetic algorithm
Menasce et al. (2007) [13]	Integer programming
Sun et al. (2007) [18]	AHP and the BrownGibson (BG) methods
Wang et al. (2007) [22]	Fuzzy-based UDDI with QoS support
Wang et al. (2010) [23]	Fuzzy linear programming
Wu and Chang (2007) [25]	QoS meta-model as the basis for the QoS and AHP modelling

Table 2.2: Overview of QoS characteristics used in web service selection.

Source	Execution					Security			Strategic				
	Response time	Accessibility	Compliance	Successability	Availability	Encryption	Authentification	Access control	Cost	Reputation	Organizational arrangement	Payment method	Monitoring
Badr et al. (2008) [1]	x	x	x	x	x	x	x	x	x	x	x	x	x
Canfora et al. (2008) [5]	x	x			x				x				
Diamadopoulou et al. (2008) [7]	x	x			x		x	x					
Lin et al. (2008) [11]	x								x				
Tran et al. (2009) [19]	x	x		x	x				x		x	x	
Wang et al. (2007) [22]	x	x		x	x								

2. Literature review. Performance of composite applications directly depends upon performance of constituent web services and efficiency of their composition. The selection of appropriate web services has been an active research area.

A number of web service selection methods have been elaborated and several typical QoS measurements used in the web service selection can be identified by analyzing these methods. Table 2.1 surveys selected web service selection methods. All these methods are multi-criteria selection methods because the web service selection is an essentially multi-criteria problem. Analytical Hierarchical Process (AHP) is the most frequently method used. It is often used together with other methods. Different methods from the artificial intelligence domain such as fuzzy algorithms and artificial neural networks are also frequently considered to account for factors, which are difficult to express analytically.

The literature review suggests that there are two main categories of attributes used in the web services selection: QoS properties and business properties category [1]. The QoS properties category may be divided into two sub categories: execution and security properties. Table 2.2 lists nonfunctional QoS characteristics considered in selected papers. Response time, accessibility and availability are the most universally used characteristics in the QoS properties category. Cost is the most frequently used business related characteristic.

However, service consumers are equally concerned about both functional and nonfunctional characteristics of services and there have been attempts to expand the QoS concept in the case of web service selection by defining it as "the degree to which a system, component or process meets customer or user needs or expectations" [9]. This definition includes evaluation of both functional and nonfunctional requirements. Unfortunately, formal evaluation of functional characteristics in the framework of web service selection is more difficult than evaluation of nonfunctional characteristics. A functional quality of service approach [10] uses similarity measures to identify interoperable web services. A QoS-aware service selection algorithm includes functional requirements in the model though these are represented only by a binary variable indicating either complete satisfaction or complete dissatisfaction of the requirement [20]. Generally, evaluation of functional characteristics either involves expert judgement or has limited resolution. Additionally, the service selection is an inherently multi-objective problem. Preemptive optimization and weighting based approaches are usually used to account for different often contradicting objectives. However, these methods again rely on judgemental appraisal of relative importance of each selection criterion. In this paper to account for different factors and objectives, an approach of expressing impact of all factors in terms of costs is used as suggested in [2].

Recently, it has been acknowledge that complexity of the service selection problem is increasing and more comprehensive service selection methods are needed. For example Vescoukis et al. (2012) [21] develop a decision support system for the service evaluation to managed environmental crises.

3. Evaluation platfom. Lifecycle of service-oriented and composite applications includes modelling, assembly, deployment and monitoring phases [24]. The evaluation platform elaborated is intended for addressing composite application design issues during the modelling and assembling phases. It has three main purposes:

1. Selection and composition of appropriate services used in design of the composite application;
2. Prediction of performance of the composite application;
3. Determination of performance requirements towards the composite application's deployment environment.

The main principles used to elaborate the evaluation platform are determined by the nature of the services selection and composition problem and the need to reduce the evaluation process overhead. In order to address the former issue, models capable to account for multiple objectives and uncertainty are used. To deal with the latter issue, model transformation and information reuse are utilized.

Figure 3.1 shows the main components of the evaluation platform. It is assumed that there are a number of candidate services proving functions required by the composite application and QoS data are available for these services. The evaluation of candidate services is performed using the optimization and simulation models. An optimization model in a form of mathematical programming model is formulated and solved using the optimization module. The optimization model selects services, which satisfy the functional requirements and have the optimal non-functional characteristics. A business process model using the Business Process modelling Notation (BPMN) notation shows composition of the services selected by the optimization model. QoS data are also represented in the business process model. The BPMN model supplemented with parameters specific to simulation purposes can be simulation using the simulation platform in order to evaluate performance of the composite application. The BPMN model can be transformed into an executable business process model (e.g., Business Process Execution Language (BPEL) model), where links with actual web services used during the execution are established. The final BPEL model is loaded into an execution platform, and the composite application is executed. The execution platform can be provided as a cloud based service. The composite application evaluation process using the proposed evaluation platform is shown in Figure 3.2. The service selection is performed jointly using the optimization and simulation model following principles of the hybrid simulation based optimization approach [6]. This approach utilizes the strength of optimization to evaluate a large number of possible service combinations and the ability of simulation to evaluate impact of stochastic factors what is important in case of using remote services. That allows for comprehensive evaluation of the selected services and their composition. If simulated performance of the composite application is not satisfactory, the evaluation process is repeated by changing candidate services, their composition or other parameters of the composite application and evaluation models.

A BPMN business process model is used as the main method for defining the composite application. It is capable of representing information required for simulation purposes. It can be used by the simulation platform and can be transformed into an executable BPEL model, which serves as a basis for implementation of the composite application. Using the transformations from the BPMN business process model to the simulation model and from the BPMN business process model to the executable BPEL model helps to reduce overhead associate with development of different evaluation models.

4. Evaluation models. The quantitative evaluation is performed using optimization and simulation models. The particular formulation of these models is case dependent though the main parameters and decision variables are common across multiple quantitative models used in design and evaluation of composite applications. The evaluation platform can be used together with different types of optimization and simulation

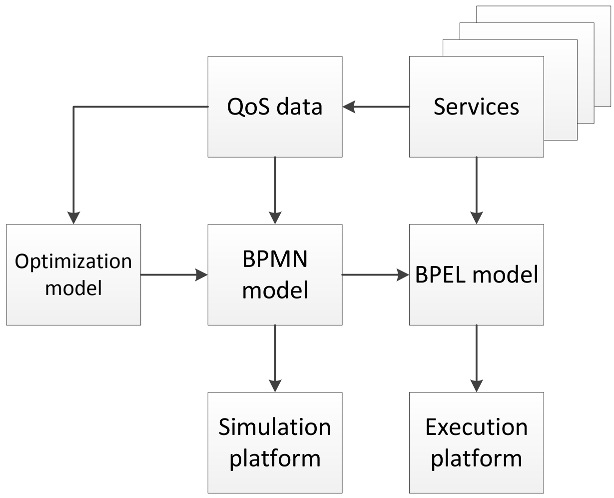

Fig. 3.1: The evaluation platform

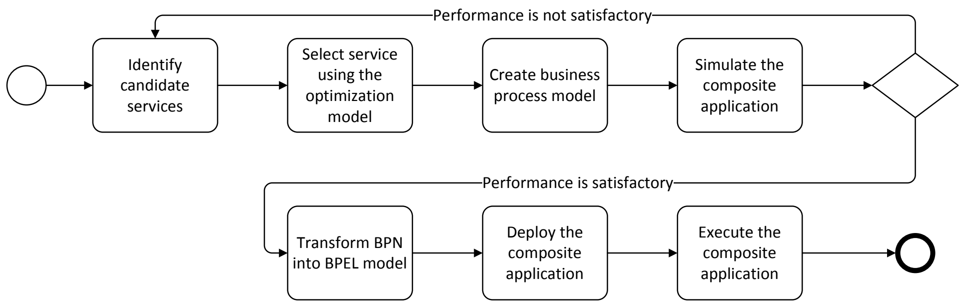

Fig. 3.2: The evaluation process

models.

4.1. Mathematical programming model. The mathematical programing model selects the most appropriate services for development of the composite application. It should account for both functional and non-functional requirements as well as to take into account multiple selection criteria. To achieve that, similarly as in [2] all selection criteria are expressed in terms of costs. These costs represent expenses associated with using the services selected from both functional and non-functional perspective. The following assumptions are made about features of the composite application:

- user requests are of different types depending upon input data provided;
- a number of candidate services provide similar functionality;
- all services can processes all types of the user requests though some of the services might need additional post-processing for some of the requests;
- if service returns an error, it is required and a positive response is received;
- if service is down for some time periods then the user requests are allocated to another service;
- each selected service incurs fixed costs (e.g., service integration costs, maintenance costs, usage fees).

The model objective function minimizes the total cost (TC) of using the selected web services over a definite planning horizon. The total cost is composed of the cost associated with service response time, the cost associate with requiring the service because of response errors and fixed costs due to using the selected web service (e.g., integration costs, maintenance cost, usage cost). Notations used to define the mathematical model are given in Table 4.1. The objective function 4.1 consists of four cost terms. The first term represents costs (denoted C_1)

Table 4.1: Notation

Notation	Description
i	index used to identify a service
j	index used to identify type of user request
N	number of candidate services
M	number of request types
$S_i \in \{0,1\}$	a decision variable indicating whether service is selected or not
X_{ij}	number of request of type jth assigned to ith service
r_j	number of request of type jth
t_{ij}	post-processing time for ith service for request of type jth
q_i^1	response time for ith service
q_i^2	percentage of requests returning an error for ith service
q_i^3	percentage of uptime for ith service
c^T	hourly composite application operating cost
c_i^F	fixed cost of using ith service
P	a large number

due to time spent on receiving responses from the selected web services, for instance, a user of the composite application who is paid an hourly rate waits till the response is received. The second term represents costs (C_2) due to the time spent on requerying services returning an error. The third term represents costs (C_3) due to time spent on post-processing of the results returned. The fourth term represents fixed costs (C_4) for using the selected services. The objective function is minimized by finding the optimal values of $\mathbf{S} = (S_1, ..., S_N)$ and $\mathbf{X} = (X_{11}, ..., X_{1M}, ..., X_{NM})$.

$$\begin{aligned} TC(\mathbf{S}, \mathbf{X}) = \sum_{i=1}^{N}\sum_{j=1}^{M} c^T q_i^1 X_{ij} + \sum_{i=1}^{N}\sum_{j=1}^{M} c^T q_i^1 q_i^2 X_{ij} \\ + \sum_{i=1}^{N}\sum_{j=1}^{M} c^T t_{ij} X_{ij} + \sum_{i=1}^{N} c_i^F S_i \to min \end{aligned} \tag{4.1}$$

$$\sum_{i=1}^{N} X_{ij} = r_j, \forall j \tag{4.2}$$

$$\sum_{j=1}^{M} X_{ij} = PS_i, \forall i \tag{4.3}$$

$$\sum_{j=1}^{M} X_{ij} \leq \sum_{j=1}^{M} q_i^3 r_j, \forall i \tag{4.4}$$

Eq. 4.2 implies that all user requests should be satisfied. Eq. 4.3 imposes that the requests can be assigned only to the services included in the composite application. Eq. 4.4 represents that a fraction of the user request cannot be met due to the service downtime if its reliability is less than one. As the result multiple services should be selected to provide a backup in the case of service unavailability. On the other hand requerying due to response errors in represented directly in the second term of the objective function. This representation of service downtime is simplified though more advance representation of this factor could make the optimization model intractable.

4.2. Simulation model. Simulation modelling is used to evaluate the composite application subject to dynamic and stochastic factors. In this case, simulation is performed using business process modelling tools, which usually have fewer simulation features than general purpose discrete event simulation tools [3] while provide a more business user friendly modelling environment and a set of concepts relevant to information

Table 5.1: List of candidate services and their properties.

Service	Geocoding by address	Geocoding by point of interest	Geocoding by intersection	q_i^1, s	q_i^2, %	q_i^3, %	c_i^F
Service 1	+	-	+	0.30	1	100	1000
Service 2	+	-	-	0.70	5	100	1000
Service 3	+	-	-	1.00	0	90	1500
Service 4	+	-	-	1,2	0	95	1800
Service 5	-	+	+	0.70	5	100	1000
Service 6	+	+	-	1.00	1	99	800

systems development [16]. In the case of composite applications, the use of business process modelling based simulators is also preferential because of their compatibility with BPEL or other executable business processes. In order to represent the composite applications and uncertainties associate with using external services, the required simulation modelling features are:

- representation of stochastically arriving user requests initiating the process execution;
- representation of stochastic service invocation response time;
- representation of random service invocation response errors and service downtime.

These features are supported by majority of business process simulation tools such as IBM Business Modeler and iGrafx Process. These tools also support simple mechanisms for allocating user requests to appropriate services though more advanced allocation mechanisms should be custom-coded (e.g., rerouting of the request during the service downtime).

5. Application example. Application of the evaluation platform is demonstrated using an example of taxi ordering call center. The company receives customer requests for taxi services. The customers order taxi by referencing their address, point-of-interest or intersection (these define the type of customer request). Call center operators lookup the particular location and identify available taxis using web services. Functionality and QoS characteristics of the candidate web services are given in Table 5.1 (these are real-life public web services though they are not named because their characteristics change continuously and exact data might not be valid at the time of publication). The table shows that, for example, Service 1 is able to geocode locations referenced by an address or a point-of-interest. Upon receiving the location information from web services, data post-processing is required. If accurate information is returned by the web service then post-processing is shorter and only includes confirmation of the customer request. However, if inaccurate information is returned (i.e., the service supports location search but not by the particular type of customer request) then post-processing takes longer and also includes manual checking using map services.

A composite application is developed to fuse results given by different web services and to minimize time operators spend on locating customers and assigning taxies to customer requests. The evaluation platform is used to identify appropriate web services and to evaluate expected performance of the composite application. Three scenarios are experimentally evaluated:

1. Standard scenario (S1) using a list of actual web services and their real-life functional and QoS characteristics;
2. Scenario with dedicated services (S2) each service is able to process only a specific type of customer requests;
3. Scenario with unreliable services (S3) service reliability is reduced to only 90% to evaluate the composite application in the case of network failure.

For the first scenario, it also important to investigate dynamical properties of the composite applications since number of customer requests and responsiveness of web services varies throughout the day. The scenarios are evaluated to determine the total cost of operating the composite application and to determine the customer request processing time. Initially, the optimization model is used to select appropriate services out of the candidate services. The optimization is performed for 1,750,000 user request made over one year. One half of the requests are by address, one third is by point of interest and one sixth is by intersection. The selected services satisfy all functional requirements and minimized the total ownership cost. The total cost breakdown for all three scenarios is given in Figure 5.1 ($c^T = 5$). Two services are selected for the first and the third scenarios, while three services are selected for the second scenario. That leads to increasing fixed costs in the case of the second scenario. The post-processing cost has the largest share since any manual operations are much more time consuming than automated service calls. The optimization model gives the same result for the first and the second scenario because optimization model has limited means to represent impact of downtime.

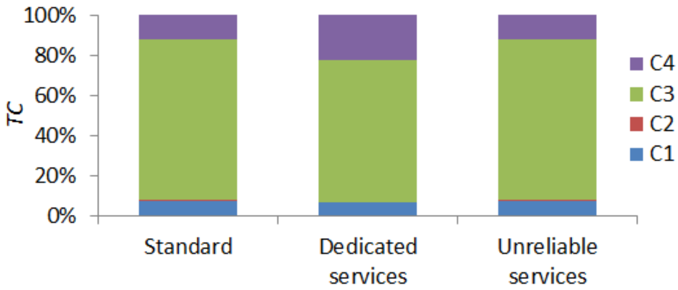

Fig. 5.1: The total cost breakdown

According to the optimization results, a BPMN business process model underlying the composite application is developed (Figure 5.2). The process starts with request receive activity representing a service for registering the user request and assigning request to a particular service depending upon the request type. The service also checks whether the service chosen is not unavailable. If the service is unavailable the request is reassigned to another service what might lead to increasing post-processing time. The appropriate location services are invoked and request post-processing is performed.

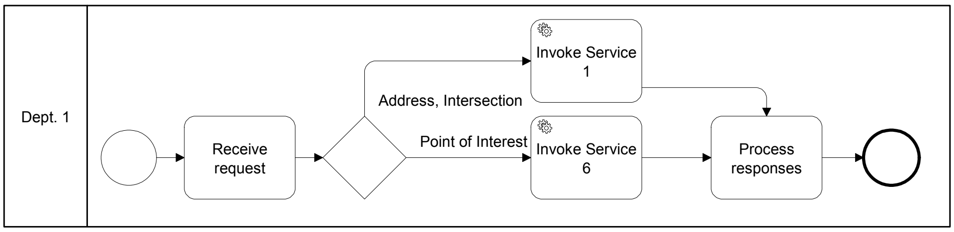

Fig. 5.2: Business process model underlying the composite application

The business process model is supplemented with data necessary for performing simulation based evaluation of the composite application. Randomly distributed execution time is specified for each activity and random service downtimes are also modeled. The customer requests are modeled as entities arriving at randomly distributed discrete time moments. Two cases are considered: 1) arrival rate is constant (R1); and 2) arrival rate varies throughout the day (R2). The case R2 represents the actual empirically observed customer requests arrival distribution. In the second case, a variable service response time is also used following the response time patterns identified by [26]. These patterns show that the service response time also exhibits the hourly variations. Therefore, impact of changes in customer requests and response time can be dynamically evaluated. Performance of the composite application is measured by a cycle time, i.e. process execution time from receiving the request till the final response to customer, and by interarrival rate of customer requests posted to external services. The latter measure is important to identify possibilities of clogging the external service.

Figure 5.3 shows a histogram of cycle time distribution for selected cases evaluated using simulation. The average cycle time for the three cases evaluated are 15.7, 17 and 16.7, respectively. These differences are statistically significant. The cycle time is the most predictable in the of uniform arrival rate of the customer request. The variable pattern of the customer requests, what is not account for in the optimization model, leads to less predictable and stable cycle times. The cycle time increase due to the service downtime also was not accounted for in the optimization model. Particularly, there are a number of requests with twice as long cycle time due to unavailability of the most appropriate service. Feeding back these results into the optimization model might result in selection of additional back-up services. Although the cycle time differences are numerically small these might lead to a necessity to higher more operators at the taxi call center over the long planning horizon.

Figure 5.4 shows interarrival time between subsequent customer requests. It can be observed that in the case of the variable customer requests pattern, there are more occasions with a short interarrival period. This particular composite application does not create a large load on external services but for other applications this result cloud be important to identify requirements for the execution platform concerning a number of simultaneous requests it is able to process.

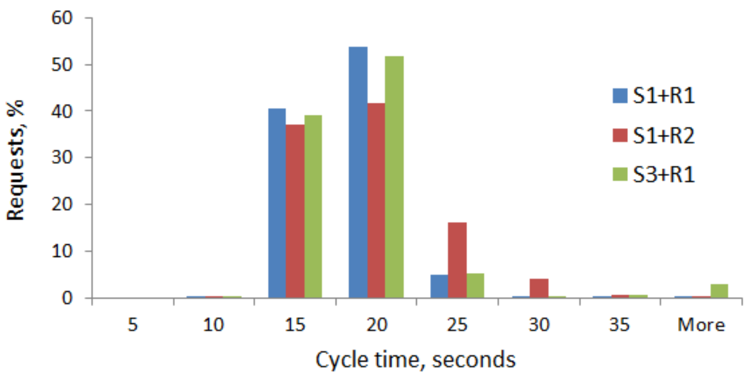

Fig. 5.3: The simulated cycle time of the composite application execution

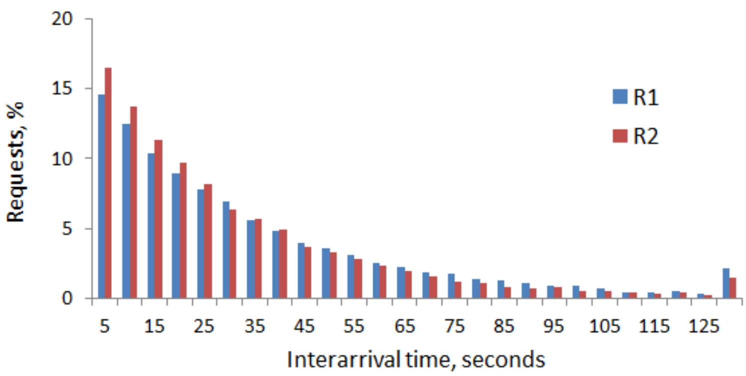

Fig. 5.4: Interarrival time between subsequent customer requests for scenario S1

For the standard scenario, performance of the composite application for demand pattern R2 is also investigated. Figure 5.5 shows the relative response time increase according to the hour of the day as suggested in [26], the actually observed number of customer requests and cycle time of the customer request processing by the composite application. It can be observed that the cycle time strongly correlated with the performance of the composite application. The number of customer requests does not have impact on the response time since service workload created by this single composite application is negligible with the global workload. However, it can be observed that, especially in the late afternoon, the increase of customer requests coincides with deteriorating service response time performance. As the result, the composite application gives the worst performance exactly when it is most frequently utilized.

In order to obtain the aforementioned results, a prototype of the simulation based evaluation platform was developed. IBM Rational System Architect is used as the core component of the platform. It is used to define all concepts relevant to development of the composite application, to develop the business process model and to

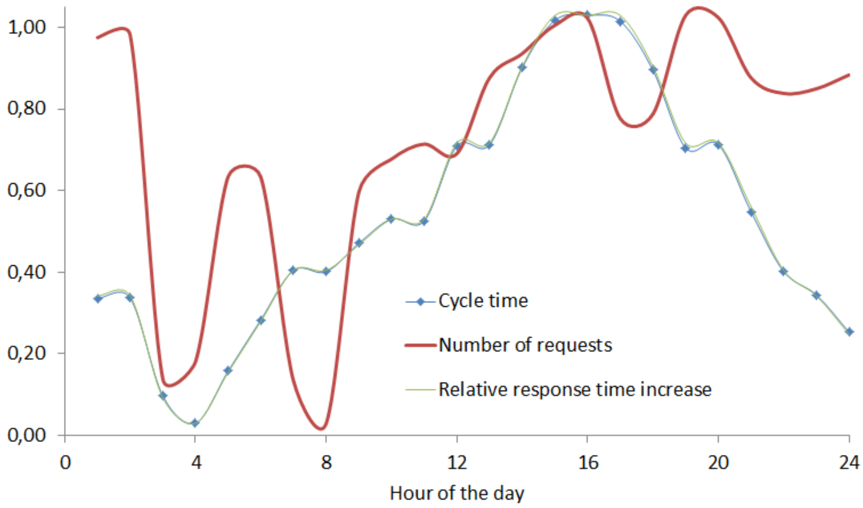

Fig. 5.5: Hourly variations of performance of the composite application. All measures are scaled to vary from 0 to 1.

perform business process simulation using the built-in Witness simulator. The optimization is performed using Lingo Solver. Executable business processes are handled using IBM Business Process Manager, which imports the business process model from IBM Rational System Architect. Data exchange between different models is performed using spreadsheet tools.

6. Conclusion. A simulation platform for development and evaluation of composite applications has been elaborated in this paper. It supports development of multiple interlinked models enabling for comprehensive evaluation of the composite applications. The experimental results show that the platform is particularly valuable to evaluate dynamic and stochastic features of the composite applications. These features cannot be effectively evaluated by just using optimization models because they become computationally intractable. The simulation results also can be used to set requirements for application execution environment. For example, the dependence of the cycle time on variable customer requests arrival pattern sets requirements for scalability in the cloud environment, where the cloud services provider should ensure that the service quality does not deteriorate at the time periods crucial for businesses support by the composite application. The obtained results are significant because without using the platform the effort of evaluation of candidate services would be much more significant and using just a single optimization model without the simulation model would not allow to fully appraise uncertainty of using internet based services in development of composite applications. The current business process modelling tools do not provide an adequate support for experimenting with business process models. In the platform prototype these functions are implemented using spreadsheets. As indicated in the literature review different types of web service selection models are available. The optimization model used in this paper can be replaced with another service selection model if appropriate, and the platform can still be used to for evaluation of the web services selected.

The composite applications are fully fledged applications including user interface and persistent data storage. The platform currently focuses on the process composition, and evaluation of other parts of the composite applications is a subject for future research.

REFERENCES

[1] Badr, Y., Abraham, A., Biennier, F. & Grosan C., *Enhancing Web Service Selection by User Preferences of Non-Functional Features*, Proceedings of the 2008 4th International Conference on Next Generation Web Services Practices (2008), pp. 60-65.

[2] Bonders, M., Grabis, J. & Kampars, J., *Combining Functional and Nonfunctional Attributes for Cost Driven Web Service Selection*, in Frontiers in Artificial Intelligence and Applications, Barzdins, J., Kirikova, M. (eds) 224 (2011), 227-239.

[3] Bosilj-Vuksic, V., Ceric, V., & Hlupic, V., *Criteria for the evaluation of business process simulation tools*, Interdisciplinary Journal of Information, Knowledge, and Management, 2(2007), 73-88.

[4] Cai, H., Hu, X., L, Q., & Cao, Q., *A novel intelligent service selection algorithm and application for ubiquitous web services environment*, Expert Systems with Applications, 36(2009), pp.2200-2212.

[5] Canfora, G., Di Penta, M., Esposito, R., & Villani, M. L., *A framework for QoS-aware binding and re-binding of composite web services*, Journal of Systems and Software, 81(2008), 1754-1769.

[6] Chandra, C. & Grabis, J., *Supply Chain Configuration: Concepts, Solutions, and Applications*, Springer: New York, 2008.

[7] Diamadopoulou, V., Makris, C., Panagis, Y., & Sakkopoulos, E., *Techniques to support web service selection and consumption with QoS characteristics*, Journal of Network and Computer Applications, 31(2008), pp.108-130.

[8] Hou, J., & Su, D., *Integration of web services technology with business models within the total product design process for supplier selection*, Computers in Industry, 57(2006), pp. 797-808.

[9] Huang, A. F. M., Lan, C., & Yang, S. J. H., *An optimal QoS-based web service selection scheme*, Information Sciences, 179(2009), pp. 3309-3322.

[10] Jeong, B., Cho, H., & Lee, C., *On the functional quality of service (FQoS) to discover and compose interoperable web services*, Expert Systems with Applications, 36(2009), pp. 5411-5418.

[11] Lin, W., Lo, C., Chao, K, & Younas, M., *Consumer-centric QoS-aware selection of web services*, Journal of Computer and System Sciences, 74(2008), pp. 211-231.

[12] Ma, Y., & Zhang, C., *Quick convergence of genetic algorithm for QoS-driven web service selection*, Computer Networks, 52(2008), pp. 1093-1104.

[13] Menasc, D. A., Ruan, H., & Gomaa, H., *QoS management in service-oriented architectures*, Performance Evaluation, 64(2007), pp. 646-663.

[14] Nolan, R.L. & Sovereign, M.G., *A recursive optimization and simulation approach to analysis with an application to transportation systems*, Management Science, 18(1972), pp. 676-690.

[15] Papazoglou, M. P., Traverso, P., Dustdar, S., & Leymann, F., *Service-oriented computing: State of the art and research challenges*, Computer, 40(2007), pp. 38-45.

[16] Ren, C., Wang, W., Dong, J., Ding, H., Shao, B. & Wang, Q., *Towards a flexible business process modelling and simulation environment*, Proceedings - Winter Simulation Conference (2008) pp. 1694.

[17] Strunk, A., *QoS-aware service composition: A survey*, Proceedings of the 8th IEEE European Conference on Web Services, ECOWS 2010 (2010), pp. 67-74.

[18] Sun, Y., He, S., & Leu, J. Y., *Syndicating web services: A QoS and user-driven approach*, Decision Support Systems, 43(2007), pp. 243-255.

[19] Tran, V. X., Tsuji, H., & Masuda, R., *A new QoS ontology and its QoS-based ranking algorithm for web services*, Simulation Modelling Practice and Theory, 17(2009), pp. 1378-1398.

[20] Tsesmetzis, D., Roussaki, I., & Sykas, E., *QoS-aware service evaluation and selection*, European Journal of Operational Research, 191(2008), pp. 1101-1112.

[21] Vescoukis, V., Doulamis, N. & Karagiorgou, S., *A service oriented architecture for decision support systems in environmental crisis management*, Future Generation Computer Systems, 28(2012), pp. 593-604.

[22] Wang, H., Lee, C., & Ho, T., *Combining subjective and objective QoS factors for personalized web service selection*, Expert Systems with Applications, 32(2007), pp. 571-584.

[23] Wang, P., Chao, K., & Lo, C., *On optimal decision for QoS-aware composite service selection*, Expert Systems with Applications, 37(2010), 440-449.

[24] Woolf, B., *Introduction to SOA governance*, IBM developerWorks, http://www.ibm.com/developerworks/library/ar-servgov/, 2007.

[25] Wu, C., & Chang, E., *Intelligent web services selection based on AHP and wiki*, Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, WI 2007, pp. 767-770.

[26] Wu, Q., Iyengar, A., Subramanian, R., Rouvellou, I., Silva-Lepe, I., & Mikalsen, T., *Combining quality of service and social information for ranking services*, 7th International Joint Conference on Service-Oriented Computing , Lecture Notes in Computer Science, Vol. 5900 (2009), pp. 561-575.

Edited by: Enn Õunapuu and Vlado Stankovski
Received: Dec 27, 2012
Accepted: Jan. 10, 2013