The Genetic Algorithm Incorporates with Rough Set Theory – An Application of the CloudERP Composition

Wen-Yau Liang

***Abstract*—Cloud computing enables many applications of Web services and rekindles the interest of providing ERP services via the Internet. This paper defines a CloudERP platform on which enterprise customers can select web services and customize a unique ERP system to meet their specific needs. A system prototype was built on the Google App Engine platform to verify the proposed composition process. Based on experimental results from running the prototype, the composition method works effectively and has great potential for supporting a fully functional CloudERP platform.**

***Index Terms*—Cloud computing, genetic algorithms, rough set theory, web services.**

I. Introduction

Several approaches and applications have been proposed to exploit the concept of Web services composition. [1] proposed a Web services-based Multidisciplinary Design Optimization (MDO) framework that synthesizes both disciplinary and cross-disciplinary resources available for MDO. Taking advantage of Web services, [2] built a biomedical digital library infrastructure called the Living Human Digital Library (LHDL) that allows clinicians and researchers to preserve, trace, and share data resources, as well as to collaborate at the data-processing level. The subscription to web services for ERP applications has two essential advantages: ease of integration and reduction in costs through the hosted application model [3]. The process of analyzing and selecting services in the Web services composition process however, are more complex than the one of analyzing and selecting parts for a product design [4]. This study proposes a method that makes use of the genetic algorithm (GA) concept and the rough set theory to solve the Web services composition problem. The GA-based method utilizes relevant knowledge extracted using the rough set theory to improve the search performance by reducing the domain range of the initial population.

II. CloudERP

Cloud computing is the delivery of a service or capability over the Internet. [5] anticipated that Cloud computing would become a key computing paradigm for the next 5 to 10 years. [6] segmented Cloud computing into three layers of service:

1) Software as a service (SaaS): Applications services delivered over the network.
2) Platform as a service (PaaS): A software development framework and components all delivered on the network. Offered as on-demand, pay for usage model.
3) Infrastructure as a service (IaaS): An integrated environment of computing resources, storage, and network fabric delivered over the network. Offered as an on-demand, pay for usage model.

The advantages of SaaS to both end users and service providers are well documented. Service providers can greatly simplify software installation and maintenance and centralizes the control of versioning. End users on the other hand can access the service “anytime, anywhere,” share data and collaborate with partners readily, while keeping their data stored safely in the infrastructure. As a result, an enterprise customer does not have to acquire the whole enterprise software suite, and yet is able to choose each module from different vendors, creating a unique, cost-efficient and customized enterprise solution [3]. The proposed composition method can be implemented as a SaaS, running on a PasS by a Cloud services provider. Fig. 1 depicts a CloudERP platform that supports interoperable service-to-service interaction over the Cloud. The CloudERP aims to provide enterprise users with the flexibility of renting an entire ERP service through multiple vendors. The platform has three major players:

1) The Cloud services provider, which enables communications among ERP providers and enterprise customers;
2) The ERP providers, which provide an XML format, computer-readable description of Web services for execution of various application functionalities;
3) The enterprise customers, which select, compose, and lease the Web services to meet their ERP objectives.

To satisfy the need of an enterprise customer for ERP application, the following platform functions have to exist and occur in the sequence as outlined in Fig. 1.

1) Submit and Assess: Evaluate each Web service submitted by ERP providers for compatibility with this platform.
2) Publish: Publish the Web service on this platform for browsers and users.
3) Select: Select and composite Web services according to the customer’s needs and ERP objectives.
4) Implement: Package the selected Web services and configure into the customer’s virtual Cloud.
5) Access: Access and run the ERP function through the virtual Cloud.

This paper focuses on the selection and composition process in step 3 and proposes a composition method with

Manuscript received October 25, 2012; revised January 7, 2013.

W. Y. Liang is with the Department of Information Management, National Changhua University of Education, Taiwan, ROC (e-mail: wyliang@cc.ncue.edu.tw).

DOI: 10.7763/IJIET.2013.V3.253

use of the GA concept and the rough set theory to select and compose web services for CloudERP users.

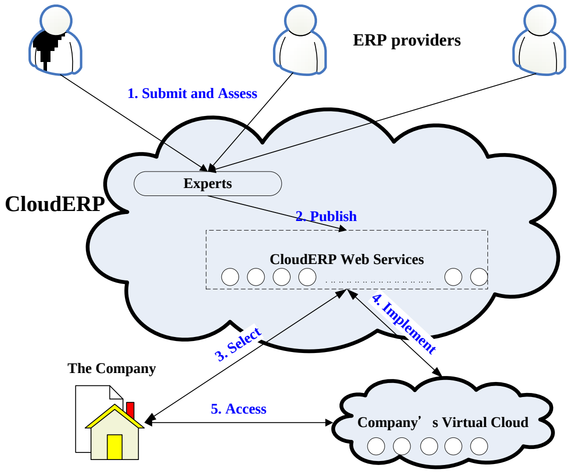

Fig. 1. A cloudERP workflow.

III. The Proposed Genetic Algorithm

This section details the proposed genetic algorithm imbedded with rough set theory. It was developed and coded with *basic units* of Web services. The *reduct* rules generated by the rough set approach were used to reduce the basic units' domain range of the initial population and validate the feasibility of offspring when a crossover is processed to achieve optimization of the objective function. It was designed to improve the effectiveness of the GA's evolution and achieve rapid convergence of the search process.

A. *The Evaluation Schema*

Deriving from expert opinion and earlier studies, [7] cited the use of criteria for ERP system selection such as total cost of ownership, functional fit of the system, user friendliness, flexibility, and vendor's reputation. Weight associated with each selected criterion is also an important factor in the decision-making process. The proposed composition method ranks composited Web services by the ranking model. Given the objective function, it considers normalizing different criteria and allowing for different weight for each criterion. The notations and computations for criterion value are specified as follows:

α, β, γ…: Represent all kinds of Web services
n: The number of Web services composition
W: Weight
C: Cost of ownership
FF: Functional fit of the ERP system
UF: User friendliness
F: Flexibility
VR: Vendor's reputation
TI Total Index

- Cost of ownership (C): C(α), C(β), C(γ) represents the cost of ownership. C(α, β, γ,…) represents an integral cost of the Web service α, β, γ….

$$C(\alpha,\beta,\gamma)=C(\alpha)+C(\beta)+C(r)+\cdots$$

- Functional fit (FF): A single Web service may influence integral composition Functional fit. FF(α), FF(β), FF(γ) represents the Functional fit of the Web service and FF$(\alpha,\beta,\gamma\cdots)$ present the integral Functional fit of the Web service α, β, γ… composition.

$$FF(\alpha,\beta,\gamma)=FF(\alpha)\times FF(\beta)\times FF(r)\times\cdots$$

- User friendliness (UF): UF(α), UF(β), UF(γ) represents the User friendliness of the Web service. UF$(\alpha,\beta,\gamma\cdots)$ represents an average User friendliness of the Web service α, β, γ….

$$UF(\alpha,\beta,\gamma)=(UF(\alpha)+UF(\beta)+UF(r)+\cdots)/n$$

- Flexibility (F): A single Web service may influence integral composition Flexibility. F(α), F(β), F(γ) represents the Flexibility of the Web service and F$(\alpha,\beta,\gamma\cdots)$ presents the integral Flexibility of the Web service α, β, γ… composition.

$$F(\alpha,\beta,\gamma)=F(\alpha)\times F(\beta)\times F(r)\times\cdots$$

- Vendor's reputation (VR): VR(α), VR(β), VR(γ) represents the VR Vendor's reputation of the Web service. VR$(\alpha,\beta,\gamma\cdots)$ represents an average Vendor's reputation of the Web service α, β, γ….

$$VR(\alpha,\beta,\gamma)=(VR(\alpha)+VR(\beta)+VR(r)+\cdots)/n$$

- Total Index (TI): Through the predefined weight (W1~W5, for each criterion) and normalization, the total index can be obtained using the following equation (*i* means the number of Web services composition).

$$TI=(\frac{C^{\max}-C}{C^{\max}-C^{\min}}\times W_1+\frac{FF-FF^{\min}_{i}}{FF^{\max}-FF^{\min}}\times W_2+\frac{UF-UF^{\min}}{UF^{\max}-UF^{\min}}\times W_3+\frac{F-F^{\min}}{F^{\max}-F^{\min}}\times W_4+\frac{VR-VR^{\min}}{VR^{\max}-VR^{\min}}\times W_5),\ \mathrm{W}_1\sim\mathrm{W}_5\ \in\ [0,1]\ \text{且}\sum_{j=1}^{5}\mathrm{W}_j=1$$

B. *The Proposed Composition Process*

The rough set routine performs off-line, and only when the database is updated. The *reduct rules* are applied in the GA's evolution process in real time while composing Web services, in attempt to increase the effectiveness of the GA's evolution, and more rapidly achieve convergence. The proposed composition process consists of two phases. The first phase is called predispose of rough set. It is a pre-composition activity and is performed off line periodically. The second phase is the rough set theory imbedded GA for web service composition.

Phase I: Predispose of Rough Set: This phase focuses on using the database to find relevant rules. The rule identification algorithm developed by [8] is used to identify and compose candidate reduct rules. The algorithm is outlined as the following five steps:

Step1: Create basic units and enter them in the database.

Step2: Calculate the lower and upper approximations for basic units.

Step3: Identify the core and reduct of attributes.

Step4: Identify the core and reduct of attributive values.

Step5: Identify the relevant rules.

Phase II: Genetic Algorithm for Web Services Composition: The reduct rules generated in phase I are applied in this phase to enhance the effectiveness of the GA evolution. This method consists of four steps as detailed below.

Step 1 (Define the parameters)

1.1: Set the initial population size to m.

1.2: Set the generation k to 1.

1.3: Set the number of chromosome to n (the number of composited Web services).

1.4: Set the mutation rate, crossover rate and termination condition.

1.5: Set weights.

1.6: Select the rules.

1.7: Set the percentage of rule matching.

Step 2 (Initialization): this initialization step is intended to produce m initial populations that satisfy the relevant constraints.

2.1: Set chromosome p to 1.

2.2: Set gene q to 1.

2.3: Select one Web service that satisfies the rule from the database.

2.4: set q=q+1.

2.5: If q<=n, then go to Step 2.3.

2.6: set p=p+1.

2.7: If p<=m, then go to Step 2.2; otherwise, go to step 3.3.

Step 3 (Evolution): the GA evolution occurs in this step.

3.1: Calculate each chromosome using the objective function.

3.2: Check termination condition. If satisfied, go to Step 4.

3.3: Use a suitable selection strategy to select the new population.

3.4: Crossover: perform crossover according to the crossover rate.

3.5: Mutation: perform the mutation process according to the mutation rate.

3.6: Set generation k= k+1, and go back to Step 3.1.

Step 4 (Detailed design)

Allow the user to evaluate the result and modulate the parameters.

IV. System Prototyping and Experimental Analysis

Many well known IT companies such as Google, Amazon, Yahoo, Microsoft, IBM, and SAP are active participants in cloud computing, on both SaaS and PaaS. Google operates a cloud computing business platform called Google App Engine (GAE), which is viewed as currently leading and more mature cloud computing platform. Software developers are able to write applications on the platform, and enterprise customers are able to customize network services [9]. The prototyping system was coded in JAVA and JSP based on the GAE for testing and validation of the proposed method. In the IT industry, product life cycle is extremely short. Companies need to deliver new products while they have market value. The cross-functional cooperation of information systems in such an IT industry segment is believed more important than the industry segments with a longer product life cycle, in order to cope with the rapid changes in customer needs and the extremely short product life cycles [10]. A typical ERP system for such an application consists of financial, human resources, manufacturing, procurement, and distribution modules [11]. Thus the scenario is an enterprise customer would like to select these five functional modules from a CloudERP platform to customize its own ERP system.

To verify the proposed method, the results were compared with those obtained by standard GA and exhaustive enumerations. The exhaustive enumerations are executed at local machine due to the limitation of GAE (A request handler has a limited amount of time to generate and return a response to a request, typically around 60 seconds. Once the deadline has been reached, the request handler is interrupted). Although the exhaustive enumerations can find all feasible solutions and the global optimum, they are computationally costly. In this case, experimental parameters are identical to those used in the previous example except that the number of candidate Web services component in each category is set to 100. The exhaustive enumerations in this study reached the global optimal composition in 7140 seconds. Accordingly, the two methods (*i.e.*, the standard GA approach and the proposed approach) were each run 1000 times. Table I shows the experimental results. The hit rate is the percentage of the global optimum obtained from the exhaustive enumerations.

TABLE I: The Comparison of Proposed Method with Standard GA

Method Index	Enumeration	Traditional GA	Proposed method
Averaged converged generation	-	117	90.7
Averaged execution time(second)	7140	0.11	0.09
Hit rate (%)	-	17%	84%

The experimental results show that the proposed composition method is significantly better than traditional GA as summarized in Tables I. The execution time for the exhaustive enumeration method is much longer than the one for the proposed method. It indicates that application of constraints in the search process improves the solution quality. Specifically when the number of candidate Web services increased from 100 to 300, the time required for the exhaustive enumerations to find the global optimum rose exponentially to 22 days. Assuming the trend continues, exhaustive enumeration methods will not be able to find the optimal solution for a much larger number of Web services. The proposed composition method on the contrary is not subject to this limitation.

V. Conclusion

This paper proposed a CloudERP platform and outlined a method for composing web services for ERP providers and enterprise users. This paper zooms in the selection process to propose a Web services composition method for Cloud platform providers in order to automatically customize an EPR service in response to a enterprise customer's need. The proposed composition method makes use of the GA concepts and employs rules generated by the rough set. The modified GA-based composition method appears to operate effectively and promote both convergence and hit rate, according to the experiments.

ACKNOWLEDGMENT

This work was partially supported by funding from the National Science Council of the Republic of China (Grant #: NSC 99-2410-H-018-016-MY3 and NSC 100-2918-I-018-001-).

REFERENCES

[1] H. J. Lee, J. W. Lee, and J. O. Lee, "Development of Web services-based Multidisciplinary Design Optimization framework," *Advances in Engineering Software*, vol. 40, pp. 176-183, March 2009.

[2] X. Zhao , E. Liu, G. J. Clapworthy, M. Viceconti, and D. Testi, "SOA-based digital library services and composition in biomedical applications," *Computer Methods and Programs in Biomedicine*, vol. 106, pp. 219–233, June 2012.

[3] C. D. Tarantilis, C.T. Kiranoudis, and N. D. Theodorakopoulos, "A Web-based ERP system for business services and supply chain management: Application to real-world process scheduling," *European Journal of Operational Research*, vol. 187, pp. 1310-1326, June 2008.

[4] W. Y. Liang and C. C. Huang, "The generic genetic algorithm incorporates with rough set theory – An application of the web services composition," *Expert Systems with Applications*, vol. 36, pp. 5549-5556, April 2009.

[5] W. Kim, "Cloud Computing: Today and Tomorrow," *Journal of Object Technology*, vol. 8, pp. 65-72, January 2009.

[6] W. Zhang, "2-Tier Cloud Architecture with Maximized RIA and SimpleDB via Minimized REST,"*2010 2nd International Conference on Computer Engineering and Technology*, vol. 6, pp. 52-56, April 2010.

[7] E. E. Karsak and C. O. Özogul, "An integrated decision making approach for ERP system selection," *Expert Systems with Applications*, vol. 36, pp. 660-667, January 2009.

[8] C. C. Huang and T. Tseng, "Rough set approach to case-based reasoning application," *Expert Systems with Applications*, vol. 26, pp. 369–385, April 2004.

[9] H. Yin, J. Han, J. Liu, and X. Hongyun, "Development and research of multimedia courseware sharing platform based on GAE," *International Conference on E-Health Networking, Digital Ecosystems and Technologies (EDT)*, 2010, pp. 289-292.

[10] Y. F. Su and C. Yang, "A structural equation model for analyzing the impact of ERP on SCM," *Expert Systems with Applications*, vol. 37, pp. 456-469, January 2010.

[11] M. K. Chang, W. Cheung, C. H. Cheng, and J. H. Y. Yeung, "Understanding ERP system adoption from the user's perspective," *Int. J. Production Economics*, vol. 113, pp. 928-942, June 2008.

Wen-Yau Liang is a professor of Information Management at National Changhua University of Education. He received his Ph.D. from the University of Iowa. His research interests are object-oriented design, artificial intelligence, intelligent agent and electronic commerce. He has published papers in journals sponsored by various societies.