Applied Soft Computing xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A cloud computing platform for ERP applications

Q1 Chin-Sheng Chen [a], Wen-Yau Liang [b,*], Hui-Yu Hsu [b]

[a] Department of Industrial and Systems Engineering, Florida International University, Miami, USA
[b] Department of Information Management, National Changhua University of Education, Taiwan, ROC

ARTICLE INFO

Article history:
Received 13 March 2013
Received in revised form 15 October 2014
Accepted 11 November 2014
Available online xxx

Keywords:
Web services composition
ERP
Cloud computing
Genetic algorithms
Rough set theory

ABSTRACT

Cloud computing enables many applications of Web services and rekindles the interest of providing ERP services via the Internet. It has the potentials to reshape the way IT services are consumed. Recent research indicates that ERP delivered thru SaaS will outperform the traditional IT offers. However, distributing a service compared to distributing a product is more complicated because of the immateriality, the integration and the one-shot-principle referring to services. 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. The CloudERP aims to provide enterprise users with the flexibility of renting an entire ERP service through multiple vendors. This paper also addresses the challenge of composing web services and proposes a web-based solution for automating the ERP service customization process. The proposed service composition method builds on the genetic algorithm concept and incorporates with knowledge of web services extracted from the web service platform with the rough set theory. 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.

1. Introduction

Traditional business applications such as computer aided design (CAD), product data management (PDM), Computer aided manufacturing (CAM), enterprise resources planning (ERP) and manufacturing execution systems (MES) all rely on a central server and procedural software. These systems are not autonomous or flexible enough to support a dynamic business environment [34]. With the advance of Internet technology and globalization, these enterprise applications, especially ERP systems have been web-enabled, providing access to information and communications via the Internet as a part of global business strategy [16]. Along with the emerging demand for mobility and on-demand services, the development of web-based ERP systems becomes an urgent research and development issue [64].

The subscription to web services for ERP applications has two essential advantages: ease of integration and reduction in costs through the hosted application model [57]. Wu et al. [67] presented a framework for measuring the scalability of service based applications in a Cloud Computing environment and propose an assignment strategy to improve the scalability of composite Web services in terms of services productivity. Recent research indicates that ERP delivered thru SaaS will outperform the traditional IT offers as a consequence of the current economic crisis and will helps the economies to recover [21]. Although ERP is lagging behind other applications in terms of SaaS based applications there seems to be a general consensus that ERP in SaaS is gaining momentum. To grab this momentum, the four big players in the ERP systems market SAP, Oracle, Sage and Microsoft are positioning their ERP offers in SaaS model [24]. However, distributing a service compared to distributing a product is more complicated because of the immateriality, the integration and the one-shot-principle referring to services [23]. Also, the process of analyzing and selecting services in the Web services composition process is more complex than the one of analyzing and selecting parts for a product design [37]. It is further complicated by the customer's request in terms of the scope of application. One specific need is the development of efficient composition methods which evaluate and optimally integrate these possibly heterogeneous services on the Web, especially in the ERP application domain, in response to an enterprise customer's request.

Therefore, this paper proposes a CloudERP platform on which enterprise customers can select web services and customize a unique ERP system to meet their specific needs. The CloudERP aims to provide enterprise users with the flexibility of renting an entire ERP service through multiple vendors. This paper also addresses the

Q2 * Corresponding author. Tel.: +886 47232105.
E-mail address: wyliang@cc.ncue.edu.tw (W.-Y. Liang).

http://dx.doi.org/10.1016/j.asoc.2014.11.009
1568-4946/

challenge of composing web services and proposes a web-based solution for automating the ERP service customization process. 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 genetic algorithm incorporates with rough set theory to solve the web services composition problem has been discussed and applied [5,36,37]. However, these all focus on how to use rough set theory to extract rules and ignore the feature of the application domain. The novelty of the proposed method lies in the application domain (Cloud ERP).

This remainder of this paper is organized as follows: Section 2 reviews the concepts of Web-based ERP and Web services composition. In Section 3, a novel ERP platform called CloudERP is proposed. In Section 4, the proposed composition method for Web ERP services is presented. In Section 5, a system prototype is presented along with experimental data analysis and then followed by Section 6 which provides concluding remarks and summary of future research directions.

2. Literature review

2.1. Web-based ERP

ERP systems are one of the most adopted information technology (IT) solutions in organizations [2]. Because of their scale and substantial resources consumption, it is not surprising that ERP systems have been a center of focus by both researchers and practitioners [11]. The key competitive edge for every enterprise in the 21st century is in its ability to prescribe, standardize, and adapt its business activities and collaborations with customers, suppliers, partners and competitors [34]. Most ERP vendors today recognize this interoperability issue as significant and have built up Internet-enabled supply chain/logistics modules to facilitate integration with the back-end systems of supply chain partners relying on a diverse set of legacy databases, IT infrastructure and applications [52]. For example, Gollakota [15] reported a company which created kiosks with Internet and computer access and operated a web portal serving the needs of the farming industry. The portal provided information relating to farming techniques, farm business information, and general information such as weather and climate, and access to the firm's ERP system. Separately, Tarantilis et al. [57] presented a Web-based ERP system developed to address business problems and manage real-world business processes ranging from a simple office automation procedure to complex supply chain planning. Zhang et al. [75] explored the IT service innovation in textile industrial clusters from a service system perspective. They argued that the IT enabled producer service could be used to ensure the structural upgrading of the textile industrial clusters. Mital et al. [44] also developed an integrative framework to identify the determinants of choice of SaaS in the specific context of SaaS based e-procurement and ERP.

In summary, one of the most important trends in the recent years is cloud computing. It has the potentials to reshape the way IT services are consumed. More recently, some ERP vendors have moved some of their offerings to the cloud e.g., SAP By Design. However, there is still a lot to be done in order for the customers to see more and more services and suites moving to the cloud. Therefore, more research efforts are still needed in order to elucidate knowledge on the marriage of the two [11].

2.2. Web services composition

The capability of composition is an important strength of any Web services provider. Rather than accessing a single service, composing services is essential as it adds better benefits to its users [13]. By ensuring high-level interoperability, Web services offer have the capability of composing compatible processes referred to as composite Web services, independent of specific platforms and computing paradigms [42]. While elementary Web services do not rely on other Web services to fulfil external requests, composite services integrate multiple service components to fulfil a customer's request [43].

Several approaches and applications have been proposed to exploit the concept of Web services composition. One research looked into the role of policies and context for framing the composition of Web services. Policies are to govern the behavior of Web services engaged in composition; and context is to support the development of adaptable Web services [42]. Yu et al. [73] designed a broker-based architecture for selecting Quality of Service (QoS)-based services. The objective of service selection is to maximize an application-specific utility function under the end-to-end QoS constraints. Park [46] presented a decentralized protocol design called the Web services co-allocation protocol, aiming to facilitate the execution of composite Web services, while Lee et al. [32] 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, Zhao et al. [77] 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. Recently, Yahyaoui et al. [70] proposed a novel matchmaking approach between fuzzy user queries and real world Web services. The matchmaking spans over a domain dependent classification step that produces fuzzy classification rules for Web services. Furthermore, these rules are leveraged to classify Web services into categories, which allow reducing the matchmaking space. One study developed an efficient approach for automatic composition of Web services using the state-of-the-art Artificial Intelligence (AI) planners [79].

Rajeswari et al. [51] revealed various challenges in the QoS parameter for Web service composition because it is difficult to recognize. In summary, Web services composition is a complex issue. The complexity initially arises from the diversity and compatibility of the composition components of Web services. It is further complicated by the customer's request in terms of the scope of application. In theory, service components are developed by different organizations and offered by different providers at different rates. There is a general need for developing principles and methodologies for managing composite Web services. One specific need is the development of efficient composition methods which evaluate and optimally integrate these possibly heterogeneous services on the Web, especially in the Cloud ERP application domain, in response to an enterprise customer's request.

3. CloudERP

Cloud computing is defined as both the applications delivered as services and the hardware and systems software in the data centers that provide those services [3]. Kim [29] anticipated that Cloud computing would become a key computing paradigm for the next 5–10 years. Cloud services can be viewed as a cluster of service solutions based on cloud computing, which involves making computing, data storage, and software services available via the Internet. Generally, cloud services can be divided into three categories [69]:

(1) Software as a service (SaaS): Applications services delivered over the network. SaaS simplifies the utilization of a large

Please cite this article in press as: C.-S. Chen, et al., A cloud computing platform for ERP applications, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.11.009

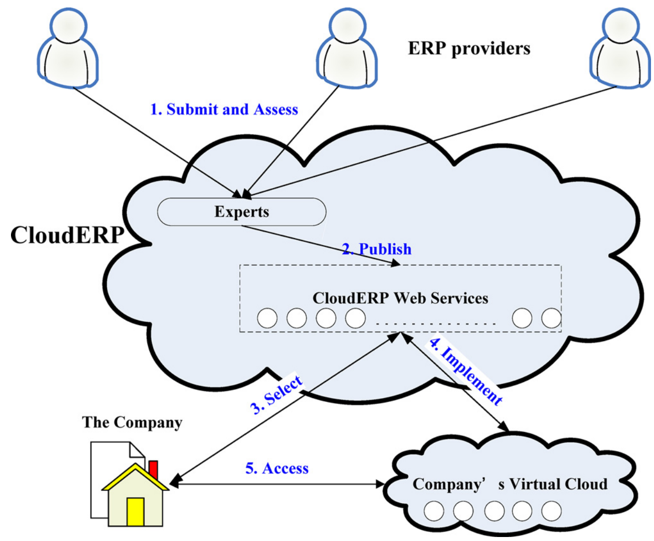

Fig. 1. A CloudERP workflow.

amount of soft- ware applications remotely, elastically and seamlessly [65].

(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. A PaaS model packages a computing platform including operating system, programming language execution environment, database, and web server. A PaaS client is able to develop and run its applications at the software layer [65].

(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.

Among them, SaaS is regarded as a potential segment and the utilization of SaaS solutions can lead to many benefits for enterprise users with profound consequences in improving IT performance [6]. 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 [57]. Within the hype of cloud services, ERP systems delivered as Software as a Service (SaaS) is receiving more focus from ERP vendors. ERP vendors have for many years developed and sold ERP as ‘standard software’ that fits the needs of many firms, and now SaaS as a new approach to deliver software has emerged [24]. The proposed composition method can be implemented as a SaaS, running on a PaaS by a Cloud services provider. Depending on its architecture, cloud computing can be categorized into three types: external/public clouds–resources dynamically provided on a self-service basis over the internet via web services from an off-site third-party provider; internal/private clouds–data and processes managed within an organization without the restriction of network bandwidth or security exposures; and hybrid clouds–the environment consisting of multiple internal and external cloud computing solutions [53]. This paper propose a hybrid clouds computing which consisting of multiple internal and external cloud computing solutions. 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 steps have to exist and occur in the sequence as outlined in Fig. 1.

Step 1 (Submit and Assess)

1.1 ERP providers submit Web services to the platform.
1.2 The platform checks the compatibility. If not, return to ERP providers.
1.3 Experts start to assess each Web services.

Step 2 (Publish)

2.1 Publish the Web service on this platform and notify users.

Step 3 (Select)

3.1 Users input requirements and constraints.
3.2 Users select composite method.

Step 4 (Implement)

4.1 The platform composites Web services and configure into the user’s virtual Cloud.
4.2 The platform notifies users.

Step 5 (Access)

5.1 Users access the composited Web service through the virtual Cloud.
5.2 Users evaluate the process and send feedbacks to the platform.

This paper focuses on the selection and composition process in step 3 and proposes a composition method with use of the GA concept and the rough set theory to select and compose web services for CloudERP users.

4. 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.

4.1. Genetic algorithm

Genetic algorithms are used to perform elegant and robust search to improve a known solution. They allow application of optimization methods to find answers in complex or poorly understood search spaces [1]. Genetic Algorithms are a powerful tool to solve combinatorial optimizing problems [41] and are a suitable choice for discredited optimization problems [28]. Vallada and Ruiz [63] proposed three genetic algorithms for the permutation flow shop scheduling problem with total tardiness minimization criterion. The results showed that the proposed algorithms were effective, outperforming the existing methods used for comparison in the study. Udhayakumar et al. [62] applied GA to solving the P-model of chance constrained data envelopment analysis (CCDEA) in a case of Indian banking sector.

GA techniques have been also applied in others fields. For example, Liang and Huang [35] used GA to solve the problem of product synthesis, where there is a conflict between performance and cost. One study proposed genetic algorithms for match-up rescheduling with non-reshuffle and reshuffle strategies which accommodate new orders by manipulating the available idle times on machines and by resequencing operations, respectively [74]. Sadrzadeh [54] presented a genetic algorithm-based meta-heuristic to solve the facility layout problem (FLP) in a manufacturing system, where the material flow pattern of the multi-line layout is considered with the multi-products. Kuo and Lin [31] proposed an evolutionary-based clustering algorithm based on a hybrid of genetic algorithm (GA) and particle swarm optimization algorithm (PSOA) for order clustering in order to reduce surface mount technology (SMT) setup time. More recently, Toledo et al. [59] applied a genetic algorithm with hierarchically structured population to solve unconstrained optimization problems. Meanwhile, one research considered the discrete optimization via simulation problem with a single stochastic constraint and presented two genetic-algorithm-based algorithms that adopt different sampling rules and searching mechanisms, and thus deliver different statistical guarantees [60]. An advanced novel heuristic algorithm is presented, the hybrid genetic algorithm with neural networks (HGA-NN), which is used to identify an optimum feature subset and to increase the classification accuracy and scalability in credit risk assessment [45].

However, typical GA techniques are viewed with shortcomings, such as poor local searching, premature converging, and slow convergence speed [25]. The next subsection thus is devoted to the genetic algorithm imbedded with the rough set theory to overcome the above search problems.

4.2. Rough set theory

Rough set theory (RST) was devised by Pawlak in 1982 as a discovery tool that can be used to induce logical patterns that are hidden in massive data. A rough set approach can capture useful information from a set of mixed data and output this information in the form of decision rules [48]. The RS approach uses a decision table with rows containing objects and columns containing criteria or features to derive decision rules through an inductive process. Rough set theory has become a well-established theory for uncertainty management in a wide variety of applications related to pattern recognition, image processing, feature selection, neural computing, conflict analysis, decision support, data mining and knowledge discovery [49]. In rough set theory, a *reduct* is defined as a minimal sufficient subset of a set of attributes, which has the same ability to discern concepts as when the full set of attributes is used [78]. Basically, the reducts represent necessary condition attributes in decision making. Further, a subset of the attributes can have more than one reduct, so simplifying the decision rules will not yield unique results. To implement the rough set theory, a procedure for determining the reducts is needed, such as generating reducts and identifying the decision rule.

The rough set theory has been applied to various areas: Tseng and Huang [61] derived the decision rules from historical data for identifying features that contribute to CRM; Chu et al. [9] proposed a expert systems for assisting mapping from performance space to design space (ESMPD); Herawan et al. [18] proposed a new technique called maximum dependency attributes (MDA) for selecting clustering attribute. The proposed approach is based on rough set theory by taking into account the dependency of attributes of the database; Shyng et al. [55] used two processes (pre process and post process) to select suitable rules and to explore the relationship among attributes; A systematic approach to analyze existing patent information based on rough set theory with the consideration of resource allocation is developed [19], a case study is presented to demonstrate the contribution of the proposed approach which assists on decision-making in patent reform or invention with constraint resource; Kaneiwa and Kudo [26] proposed a method for mining such local patterns from sequences and described an algorithm for generating decision rules that take into account local patterns for arriving at a particular decision; Wu [68] attempted to segment the ERP users into two subgroups according to the notion of Herzberg's Motivation-Hygiene theory, and further, to uncover imperative perceived benefits for distinct subgroups of ERP users employing the rough set theory. The previous literatures have shown that the rough set theory is very useful to extract knowledge and help the decision making.

Meanwhile, in the past decade, several extensions of the rough set model have been proposed to improve the effectiveness, such as the variable precision rough set model [76], the Bayesian rough set model [33], the Dominance-based rough set model [12], the fuzzy rough set model [39]. In this study, the difference of these models will not be addressed. Generally, the generated rules from each method can be applied to the proposed approach.

4.3. The evaluation schema

Many methods have been considered for ERP selection, including scoring, mathematical programming, analytic hierarchy process (AHP), and multi-criteria decision analysis [27]. Other methods such as zero–one goal programming [27], fuzzy method [66], Data envelopment analysis (DEA) [66], artificial neural network (ANN) [71], and analytic network process (ANP) [38] have also been proposed for selecting a suitable ERP.

Deriving from expert opinion and earlier studies, Karsak and Özogul [27] 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:

$\alpha,\beta,\gamma\ldots$	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): $C(\alpha)$, $C(\beta)$, $C(\gamma)$ represents the cost of ownership. $C(\alpha, \beta, \gamma, \ldots)$ represents an integral cost of the Web service $\alpha, \beta, \gamma, \ldots$.

$$C(\alpha, \beta, \gamma) = C(\alpha) + C(\beta) + (C(\gamma) + \ldots \tag{1}$$

- Functional fit (FF): A single Web service may influence integral composition Functional fit. $FF(\alpha)$, $FF(\beta)$, $FF(\gamma)$ represents the

Please cite this article in press as: C.-S. Chen, et al., A cloud computing platform for ERP applications, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.11.009

Functional fit of the Web service and $FF(\alpha, \beta, \gamma, \ldots)$ represent the integral Functional fit of the Web service $\alpha, \beta, \gamma, \ldots$ composition.

$$FF(\alpha, \beta, \gamma) = FF(\alpha) \times FF(\beta) \times FF(\gamma) \times \ldots \quad (2)$$

- User friendliness (UF): $UF(\alpha)$, $UF(\beta)$, $UF(\gamma)$ represents the User friendliness of the Web service. $UF(\alpha, \beta, \gamma \ldots)$ represents an average User friendliness of the Web service $\alpha, \beta, \gamma \ldots$.

$$UF(\alpha, \beta, \gamma) = (UF(\alpha) + UF(\beta) + UF(\gamma) + \ldots)/n \quad (3)$$

- Flexibility (F): A single Web service may influence integral composition Flexibility. $F(\alpha), F(\beta), F(\gamma)$ represents the Flexibility of the Web service and $F(\alpha, \beta, \gamma, \ldots)$ represents the integral Flexibility of the Web service $\alpha, \beta, \gamma \ldots$ composition.

$$F(\alpha, \beta, \gamma) = F(\alpha) \times F(\beta) \times F(\gamma) \times \ldots \quad (4)$$

- Vendor's reputation (VR): $VR(\alpha)$, $VR(\beta)$, $VR(\gamma)$ represents the VR Vendor's reputation of the Web service. $VR(\alpha, \beta, \gamma \ldots)$ represents an average Vendor's reputation of the Web service $\alpha, \beta, \gamma \ldots$.

$$VR(\alpha, \beta, \gamma) = (VR(\alpha) + VR(\beta) + VR(\gamma) + \ldots)/n \quad (5)$$

- 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).

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

4.4. The proposed composition process

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. GA algorithms are efficient search methods based on the principles of natural selection and population genetics in which random operators in a population of candidate solutions are employed to generate new points in the search space [8]. The increase in data and information however often hinders the performance and capacity of the GA, raising the cost of finding a solution by using it. To overcome the problem, Passone et al. [47] combined the GA with guidance provided by domain-specific knowledge. The proposed method makes use of the rough set theory introduced by Pawlak as a mathematical method [50] to improve the GA performance. With the rough set method, the minimal attribute sets can be extracted without deterioration in the quality of approximation and minimal length decision rules corresponding to lower or upper approximation [22]. 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. Both phases are outlined in Fig. 2 and detailed as follows:

Phase I: Predispose of rough set: This phase focuses on using the database to find relevant rules. The rule identification algorithm developed by Huang and Tseng [20] is used to identify and compose candidate reduct rules. The algorithm is outlined as the following five steps:

Step 1: Create basic units and enter them in the database.
Step 2: Calculate the lower and upper approximations for basic units.
Step 3: Identify the core and reduct of attributes.
Step 4: Identify the core and reduct of attributive values.
Step 5: 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.

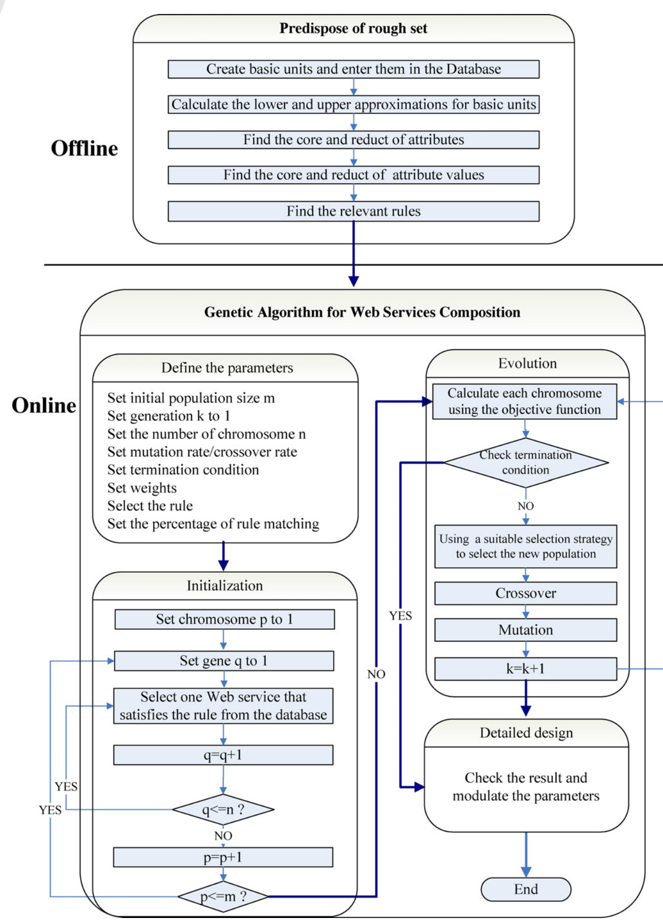

Fig. 2. The proposed web service composition method.

Please cite this article in press as: C.-S. Chen, et al., A cloud computing platform for ERP applications, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.11.009

Table 1
A partial table of the Web services component database.

Functionality	Vendor	Flexibility (range 1–100)	Reputation (range 1–10)	Cost (range 1–10)	Functional fit (range 1–100).	User friendliness (range 1–10)
Finance	SAP	98	9	8	89	7
Distribution	SAP	69	6	7	93	9
Human Resource	Salesforces	95	7	8	93	7
Manufacturing	SAP	55	9	10	97	7
Manufacturing	Microsoft	62	8	7	97	8
Finance	Microsoft	65	10	8	87	6
Procurement	Microsoft	91	7	7	91	6
Distribution	Oracle	94	6	9	90	7

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)

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

5. 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 [72].

The prototyping system was coded in JAVA and JSP based on the GAE for testing and validation of the proposed method. The implementation scenario of CloudERP is shown in Fig. 3. The information of Web services components was published and stored in the system database. Table 1 represents a partial table of Web services components (The table is set randomly). The value of each criterion is predefined by experts.

5.1. Example illustration

In the IT industry, product life cycle is extremely short. Companies need to deliver new products while they have market value. In some IT industry segments, original equipment manufacturing (OEM) and original design manufacturing (ODM) are the main business, of which companies are not involved in their customers' sales and marketing activities. 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 [56]. A typical ERP system for such an application consists of financial, human resources, manufacturing, procurement, and distribution modules [7]. Thus the scenario is an enterprise customer would like to select these five functional modules from a CouldERP platform to customize its own ERP system. This system prototype is intended to use roulette wheel selection and crossover rate/mutation rate to generate the offspring. Since this enterprise customer wants to pay more attention to flexibility, thus the weight for flexibility is set at 0.6. The weights for the other four criteria are set at 0.1. The proposed method works with the example as follows:

Phase I: Predispose of rough set

Step 1: Create basic units and enter them in the database

Following the approach proposed in Liang and Huang [37], used supplier, type, and implementation language were selected as the conditional attributes of the rough set. Also considered were network-related parameters such as network domain, download frequency, and reputation as argued by Tian et al. [58], Hansen et al. [17], and Ko et al. [30]. Consequently supplier, implementation language, network domain and number of downloads were adopted as the four conditional attributes. On the other hand, high performance, high reputation, and low cost were used as the three decision variables. Each attribute in Table 2 has its own code name. Thirty records were randomly selected from the dataset for this example. Based on Table 2, a decision table was generated as in Table 3 and then organized into an elementary set by their attributive values as shown in Table 4.

Step 2: Calculate the lower and upper approximations for basic units

The elementary sets' upper and lower approximations were computed in this step. And decision attribute's elementary set was completely classified as shown in Table 5.

Step 3: Find the core and reduct of attributes

This step created a discernibility matrix to obtain the core and reducts, using the absorption law to calculate the reduct result. In this example, the reduct {C3: No. of download; C4: Implemented Language} was as shown in Table 6.

Table 2
Description of all attributes.

Conditional attributes			Decision attribute	
C1: Supplier	C2: Network domain	C3: No. of download	C4: Implemented language	O: Outcomes
1 – Salesforces	(1) Taiwan	(1) Low (0∼15)	(1) JAVA	(1) High performance
2– SAP	(2) USA	(2) Middle (16∼30)	(2) VB	(2) Low cost
3– Microsoft	(3) India	(3) High (31∼)	(3) Delphi	(3) High reputation
4 – Oracle			(4) C# (5) ABAP	

Please cite this article in press as: C.-S. Chen, et al., A cloud computing platform for ERP applications, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.11.009

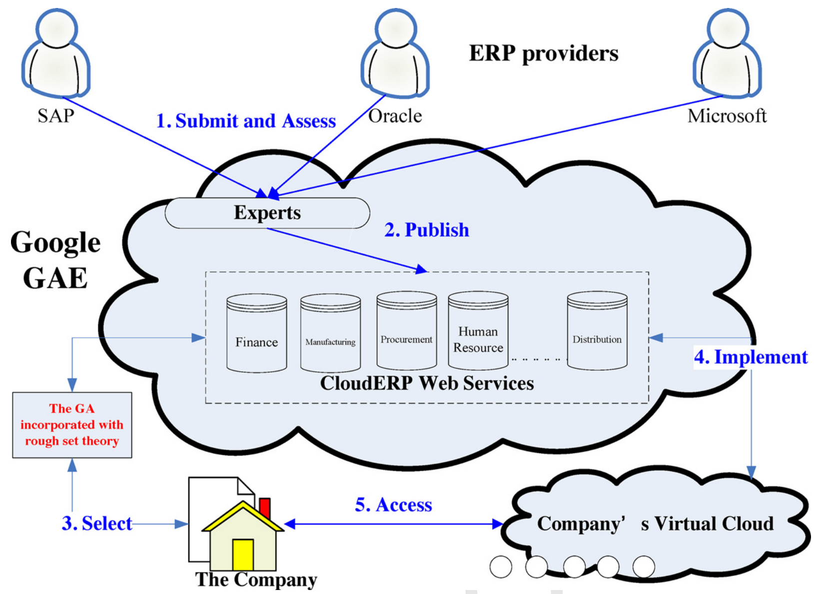

Fig. 3. The implementation scenario of CloudERP.

Table 3
Decision Table.

Object No.	C1	C2	C3	C4	Decision attributes
X1	3	3	3	1	1
X2	3	3	2	2	2
X3	2	1	3	1	1
X4	1	2	1	2	2
⋮	⋮	⋮	⋮	⋮	⋮
X27	3	2	3	1	1
X28	3	2	1	2	2
X29	3	3	3	3	3
X30	2	1	3	1	1

Table 4
Elementary set.

Cluster No.	Objects	Decision attributes
1	{x1,x3,x7,x10,x11,x13,x17,x20,x21,x23,x27,x30}	1
2	{x2,x4,x8,x12,x14,x18,x22,x24,x28}	2
3	{x5,x6,x9,x15,x16,x19,x25,x26,x29}	3

Table 5
Lower and upper approximations.

Cluster No.	Number of objects	Lower approximations	Upper approximations	Accuracy
1	12	12	12	1.0
2	9	9	9	1.0
3	9	9	9	1.0

Table 6
Reduct of attributes.

	Reduct
1	{C3, C4}

Step 4: Find the core and reduct of attributive values

The unnecessary values in the condition attributes in the decision table were eliminated in this step, according to the reduct {C3, C4} and the discernibility matrix. After calculation, the final decision table version was generated as shown in Table 7.

Step 5: Find the relevant rules

Table 7
Final decision table.

	C3	C4	O
X1	3	1	1
X2	*	2	2
X3	3	1	1
X4	*	2	2
⋮	⋮	⋮	⋮
X27	3	1	1
X28	*	2	2
X29	3	*	3
X30	3	1	1

Table 8
Decision rule.

Reduct No.	C3	C4	O
1	3	1	1
2	*	2	2
3	3	*	3

Table 9
The explanation of decision rules.

Rule No.	Meaning
1	IF the No. of download is high and iimplemented language is Java, then the Web service has high performance.
2	IF implemented language is VB, then the Web service has low cost.
3	IF the No. of download is high, then the Web service has high reputation.

In this step, a set of decision rules was summarized as shown in Table 8, in accordance with Table 7. The meaning of the decision rules is interpreted in Table 9.

Phase II: Genetic Algorithm for Web services composition

Step 1 (Define the parameters)

1.1 Set the initial population size m to 300.
1.2 Set the generation k to 1.
1.3 Set the number of chromosome n as 5 (number of composited Web services)
1.4 Set the mutation rate to 0.001, crossover rate as 0.8 and termination condition as 500 generations.

Please cite this article in press as: C.-S. Chen, et al., A cloud computing platform for ERP applications, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.11.009

Table 10
The partial of initial populations.

Chromosome No.	Category				
	A	B	C	D	E
1	65	69	80	49	30
2	15	32	96	89	68
3	39	60	16	16	56
4	24	62	60	82	8
5	7	44	90	36	92
6	69	42	89	66	40
7	36	6	26	92	89
8	87	66	27	59	80
9	47	70	70	82	49
10	42	76	26	12	9

1.5 Set the weights (Cost: 0.1; Functional Fit: 0.1; User Friendliness: 01; Flexibility: 0.6; Reputation: 0.1).
1.6 Select the rule: the final reduct rule in this example was applied to the genetic algorithm to reduce the domain range of the initial population. Since this company assigns a heavier weight to flexibility (0.6), the reduct rule 1 in Table 9 which says "IF the No. of downloads is high and implemented Language is Java, then the Web service has high performance." was selected since the extent of flexibility can be measured by one of its metrics: efficiency [14].
1.7 Set the percentage of rule matching to 90%: To avoid finding a local optimal solution, weight analysis was used to make the rules fuzzy rather than absolute. Using the weight, the reduct rule 1 in Table 9 implies that 90% of the initial population selected was considered satisfactory.

Step 2 (Initialization)
This step aims to produce 500 populations, composed of five categories (A: Finance, B: Human Resource, C: Manufacturing, D: Procurement, and E: Distribution), each of which satisfies the reduct rule. Table 10 is a partial population that was obtained.
Step 3 (Evolution)
This step focuses on the GA evolution of the 500 generations. In this case, the proposed GA converges at the 70th generation, at which all the population is the same and the fitness functions are all equal to 1. The best, worst and average fitness (objective) converged at 1 as shown in Figure 4. The evolution process therefore was terminated. This example was converged at {36,66,36,92,96}
Step 4 (Detailed design)
The final result for this case study is summarized in Table 11. The enterprise user now can adjust the parameters to test and verify.

5.2. Experimental analysis

5.2.1. Experiment I – small sample

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 s. Once the deadline has been reached, the request handler is interrupted). Although the exhaustive enumerations can find

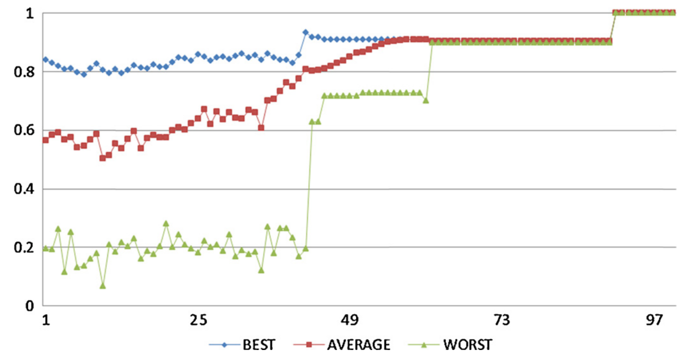

Fig. 4. The converged result of genetic evolution.

Table 12
The comparison of proposed approach with standard GA – small sample.

Index	Method		
	Enumeration	Traditional GA	Proposed approach
Averaged converged generations	–	98.2	94.1
Averaged execution time (s)	222	0.101	0.089
Hit rate (%)	–	39%	89%

Table 13
The comparison of proposed method with standard GA – large sample.

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

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 50. The exhaustive enumerations in this study reached the global optimal composition in 222 s. Accordingly, the two methods (i.e., the standard GA approach and the proposed approach) were each run 1000 times. Table 12 shows the experimental results. The hit rate is the percentage of the global optimum obtained from the exhaustive enumerations. The proposed method has a much higher high rate.

5.2.2. Experiment II – large sample

In this case, experimental parameters are identical to those used in the previous example except that the number of candidate Web services components in each category is set to 100. Table 13 shows the proposed method has a much higher hit rate.

Table 11
The final result.

Functionality	ID	Vendor	Flexibility	Reputation	Cost	Functional fit	User friendliness
Finance	36	Microsoft	99	7	4	96	7
Human Resource	66	Microsoft	99	9	6	99	8
Manufacturing	36	Microsoft	99	9	3	92	7
Procurement	92	SAP	99	8	7	94	8
Distribution	96	Microsoft	99	9	5	92	10

Please cite this article in press as: C.-S. Chen, et al., A cloud computing platform for ERP applications, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.11.009

Table 14
Hit rates among different populations.

Population No.	100	200	300
Hit rate	84.48%	85.41%	85.96%

Table 15
Hit rates among different crossover rates.

Crossover rate	0.7	0.8	0.85	0.9
hit rate	85.12%	85.36%	85.4%	85.31%

Table 16
Hit rates among different mutation rates.

Mutation rate	0.001	0.0015	0.005
Hit rate	85.25%	85.25%	85.36%

5.2.3. Experiment III – sensitivity analysis

Sensitivity analysis is often used to analyze how sensitive a system is with respect to the change of parameters [40]. A sensitivity analysis of the proposed GA method was carried out with three parameters, which are population size, crossover rate and mutation rate. These parameters largely determine the success and efficiency of a GA routine for solving a specific problem. 36 sets of parameters in total combining different crossover rates (0.7, 0.8, 0.85, 0.9), mutation rates (0.001, 0.0015, 0.005) and population numbers (100, 200, 300) were used to run the experiments and to obtain the hit rate. The findings are summarized below:

- The greater the population, the higher the hit rate as shown in Table 14.
- There is no significant difference in hit rate among different crossover rates as shown in Table 15.
- There is no significant difference in hit rate among different mutation rates as shown in Table 16.

The sensitivity analysis on the parameters shows that the user can recognize which parameter to focus on for the problem of interest. For example, the nature of the hit rate in this case study highly relates to the population number. Eiben et al. [10] conducted a comprehensive review and classification of parameter control methods for evolutionary algorithms. Yet, the selection of these control parameters is rather complex and needs further research, beyond the scope of this study.

5.3. Discussion

The experimental results show that the proposed composition method is significantly better than traditional GA as summarized in Tables 12 and 13. 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.

Johansson and Ruivo [24] explore vendor's perspective on what factors affect adopting ERP as SaaS and found 10 factors: Costs, Security, Availability, Usability, Implementation, Ubiquity, Flexibility, Compatibility, Analytics and Best-practices. Costs, data security and system availability were perceived by the experts as the most important factors in customer perspective for adopting ERP systems in a SaaS delivery model. A SaaS provider should realize that a successful market establishment of its offer lays not so much on the product itself but on the kind of support given in the SaaS model and the customer experience with provided service. That is, the paradigm changes from product feature to service trust. Given the emerging trend of ERP delivered thru CloudERP platform, researchers can investigate other issues like privacy, customer-centric and type of firm capabilities. The real strength of cloud computing is that it is a catalyst for more innovation. In fact, as cloud computing continues to become cheaper and more ubiquitous, the opportunities for combinatorial innovation will only grow. It is true that this inevitably requires more creativity and skill from IT and business executives. In the end, this not something to be avoided. It should be welcomed and embraced [4].

6. 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. Toward developing a fully functional CloudERP platform, more research and development efforts are needed to refine the proposed composition process, especially with respect to the submission, assessment, publishing, and implementation activities.

Acknowledgment

This work was partially supported by funding from the Ministry of Science and Technology of the Republic of China (NSC Q3 99-2410-H-018-016-MY3; NSC 100-2918-I-018-001; MOST 103-2410-H-018-015-MY2).

References

[1] H. Ahn, K. Kim, Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach, Appl. Soft Comput. 9 (2009) 599–607.
[2] M. Al-Mashari, A. Al-Mudimigh, M. Zairi, Enterprise resource planning: a taxonomy of critical factors, Eur. J. Oper. Res. 146 (2) (2003) 352–364.
[3] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, A view of cloud computing, Commun. ACM 53 (4) (2010) 50–58.
[4] E. Brynjolfsson, P. Hofmann, J. Jordan, Cloud computing and electricity: beyond the utility model, Commun. ACM 53 (2012) 32–34.
[5] M.M. Cariappa, M.K. Nair, Applying rough set theory to genetic algorithm for web service composition, Int. J. Comput. Sci. Inform. 2 (4) (2012) 2231–5292.
[6] D. Catteddu, G. Hogben, Cloud Computing: Benefits, Risks and Recommendations for Information Security, European Network and Information Security Agency (ENISA), 2009, pp. 1–125.
[7] M.K. Chang, W. Cheung, C.H. Cheng, J.H.Y. Yeung, Understanding ERP system adoption from the user's perspective, Int. J. Prod. Econ. 113 (2008) 928–942.
[8] T.C. Chen, T.C. Hsu, A GAs based approach for mining breast cancer pattern, Expert Syst. Appl. 30 (4) (2006) 674–681.
[9] X.Z. Chu, L. Gao, H.B. Qiu, W.D. Li, X.Y. Shao, An expert system using rough sets theory and self-organizing maps to design space exploration of complex products, Expert Syst. Appl. 37 (11) (2010) 7364–7372.
[10] A.E. Eiben, R. Hinterding, Z. Michalewicz, Parameter control in evolutionary algorithm, IEEE Trans. Evol. Comput. 3 (2) (1999) 124–141.
[11] A. Elragal, M. Haddara, The future of ERP systems: look backward before moving forward, Proc. Technol. 5 (2012) 21–30.
[12] T.F. Fan, C.J. Liau, D.R. Liu, Dominance-based fuzzy rough set analysis of uncertain and possibilistic data tables, Int. J. Approx. Reason. 52 (9) (2011) 1283–1297.
[13] C. Ghedira, H. Mezni, Through personalized web service composition specification: from BPEL to C-BPEL, Electron. Notes Theor. Comput. Sci. 146 (1) (2006) 117–132.
[14] W. Golden, P. Powell, Towards a definition of flexibility: in search of the Holy Grail, Int. J. Manage. Sci. 28 (2000) 373–384.
[15] K. Gollakota, ICT use by businesses in rural India: the case of EID Parry's Indiagriline, Int. J. Inf. Manage. 28 (2008) 336–341.

[16] M. Gupta, A. Kohli, Enterprise resource planning systems and its implications for operations function, Technovation 26 (5/6) (2006) 687–696.
[17] K.M. Hansen, K. Jonasson, H. Neukirche, An empirical study of software architectures' effect on product quality, J. Syst. Softw. 84 (2011) 1233–1243.
[18] T. Herawan, M.M. Deris, J.H. Abawajy, A rough set approach for selecting clustering attribute, Knowl. Based Syst. 23 (3) (2010) 220–231.
[19] C.C. Huang, W.Y. Liang, S.H. Lin, T.L. Tseng, H.Y. Chiang, A rough set based approach to patent development with the consideration of resource allocation, Expert Syst. Appl. 38 (3) (2011) 1980–1992.
[20] C.C. Huang, T. Tseng, Rough set approach to case-based reasoning application, Expert Syst. Appl. 26 (3) (2004) 369–385.
[21] IDC, IDC Predictions: An Economic Pressure Cooker Will Accelerate the IT industry, 2009.
[22] M. Inuiguchi, T. Miyajima, Rough set based rule induction from two decision tables, Eur. J. Oper. Res. 181 (3) (2007) 1540–1553.
[23] B. Iyer, J.C. Henderson, Preparing for the future: understanding the seven capabilities of cloud computing, MIS Q. Exec. 9 (2012) 117–131.
[24] B. Johansson, P. Ruivo, Exploring factors for adopting ERP as SaaS, Proc. Technol. 9 (2013) 94–99.
[25] B. Kadri, M. Boussahla, F.T. Bendimerad, Phase-only planar antenna array synthesis with fuzzy genetic algorithms, in: Proceedings of the CoRR, 2010.
[26] K. Kaneiwa, Y. Kudo, A sequential pattern mining algorithm using rough set theory, Int. J. Approx. Reason. 52 (6) (2011) 881–893.
[27] E.E. Karsak, C.O. Özogul, An integrated decision making approach for ERP system selection, Expert Syst. Appl. 36 (2009) 660–667.
[28] T.U. Kim, J.W. Shin, I.H. Hwang, Stacking sequence design of a composite wing under a random gust using a genetic algorithm, Comput. Struct. 85 (2007) 579–585.
[29] W. Kim, Cloud computing: today and tomorrow, J. Object Technol. 8 (2009) 65–72.
[30] J.M. Ko, C.O. Kim, I. Kwon, Quality-of-service oriented web service composition algorithm and planning architecture, J. Syst. Softw. 81 (2008) 2079–2090.
[31] R.J. Kuo, L.M. Lin, Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering, Decis. Support Syst. 49 (4) (2010) 451–462.
[32] H.J. Lee, J.W. Lee, J.O. Lee, Development of web services-based multidisciplinary design optimization framework, Adv. Eng. Softw. 40 (2009) 176–183.
[33] D.S. lezak, W. Ziarko, The investigation of the Bayesian rough set model, Int. J. Approx. Reason. 40 (2005) 81–91.
[34] B. Li, M. Li, Research and design on the refinery ERP and EERP based on SOA and the component oriented technology, in: International Conference on Networking and Digital Society, 1, 2009, pp. 85–88.
[35] W. Liang, C. Huang, A hybrid approach to constrained evolutionary computing: case of product synthesis, Omega 36 (2008) 1072–1085.
[36] W.Y. Liang, Apply rough set theory into the web services composition, in: Proceedings of the IEEE 22nd International Conference on Advanced Information Networking and Applications, Okinawa, Japan, March 25–28, 2008.
[37] W.Y. Liang, C.C. Huang, The generic genetic algorithm incorporates with rough set theory – an application of the web services, Expert Syst. Appl. 36 (3) (2009) 5549–5556.
[38] C.T. Lin, C.B. Chen, Y.C. Ting, An ERP model for supplier selection in electronics industry, Expert Syst. Appl. 38 (2011) 1760–1765.
[39] G.L. Liu, Using one axiom to characterize rough set and fuzzy rough set approximations, Inf. Sci. 223 (2013) 285–296.
[40] Y. Liu, F. Sun, Sensitivity analysis and automatic calibration of a rainfall–runoff model using multi-objectives, Ecol. Inform. 5 (2010) 304–310.
[41] Y. Ma, C. Zhang, Quick convergence of genetic algorithm for QoS-driven web service selection, Comput. Netw. 52 (2008) 1093–1104.
[42] Z. Maamar, D. Benslimane, P. Thiran, G. Ghedira, S. Dustdar, Towards a context-based multi-type policy approach for Web services composition, Data Knowl. Eng. 62 (2) (2007) 327–351.
[43] T. Madhusudan, N. Uttamsingh, A declarative approach to composing web services in dynamic environments, Decis. Support Syst. 41 (2) (2006) 325–357.
[44] M. Mital, A. Pani, R. Ramesh, Determinants of choice of semantic web based software as a service: an integrative framework in the context of e-procurement and ERP, Comput. Ind. (2014) (in press).
[45] S. Oreski, G. Oreski, Genetic algorithm-based heuristic for feature selection in credit risk assessment, Expert Syst. Appl. 41 (2014) 2052–2206.
[46] J. Park, A high performance backoff protocol for fast execution of composite web services, Comput. Ind. Eng. 51 (1) (2006) 14–25.
[47] S. Passone, P.W.H. Chung, V. Nassehi, Incorporating domain-specific knowledge into a genetic algorithm to implement case-based reasoning adaptation, Knowl. Based Syst. 19 (3) (2006) 192–201.
[48] Z. Pawlak, Rough classification, Int. J. Man Mach. Stud. 20 (5) (1984) 469–483.
[49] Y. Qian, S. Li, J. Liang, Z. Shi, F. Wang, Pessimistic rough set based decisions: a multigranulation fusion strategy, Inf. Sci. 264 (2014) 196–210.
[50] F. Questier, I. Arnaut-Rollier, B. Walczak, D.L. Massart, Application of rough set theory to feature selection for unsupervised clustering, Chemometr. Intell. Lab. Syst. 63 (2) (2002) 155–167.
[51] M. Rajeswari, G. Sambasivam, M. Balaji, S.S. Basha, T. Vengattaraman, P. Dhavachelvan, Appraisal and analysis on various web service composition approaches based on QoS factors, J. King Saud Univ. Comput. Inf. Sci. 26 (2014) 143–152.
[52] C. Ranganathan, T.S.H. Teo, J. Dhaliwal, Web-enabled supply chain management: key antecedents and performance impacts, Int. J. Inf. Manage. 31 (6) (2011) 533–545.
[53] B.P. Rimal, E. Choi, I. Lumb, A taxonomy and survey of cloud computing systems, in: Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC, 2009, pp. 44–51.
[54] A. Sadrzadeh, A genetic algorithm with the heuristic procedure to solve the multi-line layout problem, Comput. Ind. Eng. 62 (2012) 1055–1064.
[55] J.Y. Shyng, H.M. Shieh, G.H. Tzeng, An integration method combining Rough Set Theory with formal concept analysis for personal investment portfolios, Knowl. Based Syst. 23 (6) (2010) 586–597.
[56] Y.F. Su, C. Yang, A structural equation model for analyzing the impact of ERP on SCM, Expert Syst. Appl. 37 (2010) 456–469.
[57] C.D. Tarantilis, C.T. Kiranoudis, N.D. Theodorakopoulos, A Web-based ERP system for business services and supply chain management: application to real-world process scheduling, Eur. J. Oper. Res. 187 (2008) 1310–1326.
[58] M. Tian, A. Gramm, H. Ritter, J. Schiller, R. Winter, A survey of current approaches towards specification and management of quality of service for web services, Praxis der Informationsverarbeitung und Kommunikation 27 (3) (2004) 132–139.
[59] C.F.M. Toledo, L. Oliveira, P.M. França, Global optimization using a genetic algorithm with hierarchically structured population, J. Comput. Appl. Math. 261 (2014) 341–351.
[60] S.C. Tsai, S.Y. Fu, Genetic-algorithm-based simulation optimization considering a single stochastic constraint, Eur. J. Oper. Res. 236 (2014) 113–125.
[61] T.L. Tseng, C.C. Huang, Rough set-based approach to feature selection in customer relationship management, Omega 35 (2007) 365–383.
[62] A. Udhayakumar, V. Charles, M. Kumar, Stochastic simulation based genetic algorithm for chance constrained data envelopment analysis problems, Omega 39 (2011) 387–397.
[63] E. Vallada, R. Ruiz, Genetic algorithms with path relinking for the minimum tardiness permutation flow shop problem, Omega 38 (2010) 57–67.
[64] T.L. Wang, C.H. Su, P.Y. Tsai, T.Y. Liang, W.H. Wu, Development of a GridERP architecture: integration of grid computing and enterprise resources planning application, in: 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008, pp. 1–4.
[65] X. Wang, X. Xu, An interoperable solution for cloud manufacturing, Robot. Comput. Integr. Manuf. 29 (2013) 232–247.
[66] C.-C. Wei, M.-J.J. Wang, A comprehensive framework for selecting an ERP system, Int. J. Proj. Manage. 22 (2004) 161–169.
[67] J. Wu, Q. Liang, E. Bertino, Improving scalability of software cloud for composite web services, in: IEEE International Conference on Cloud Computing, 2009, pp. 143–146.
[68] W.W. Wu, Segmenting and mining the ERP users' perceived benefits using the rough set approach, Expert Syst. Appl. 38 (6) (2011) 6940–6948.
[69] W.W. Wu, L.W. Lan, Y.T. Lee, Exploring decisive factors affecting an organization's SaaS adoption: a case study, Int. J. Inf. Manage. 31 (6) (2011) 556–563.
[70] H. Yahyaoui, M. Almulla, h.S. Own, A novel non-functional matchmaking approach between fuzzy user queries and real world web services based on rough sets, Future Gener. Comput. Syst. 35 (2014) 27–38.
[71] H.R. Yazgan, S. Boran, K. Goztepe, An ERP software selection process with using artificial neural network based on analytic network process approach, Expert Syst. Appl. 36 (2009) 9214–9222.
[72] H. Yin, J. Han, J. Liu, X. Hongyun, Development and research of multimedia courseware sharing platform based on GAE, in: International Conference on E-Health Networking, Digital Ecosystems and Technologies (EDT), 2010, pp. 289–292.
[73] T. Yu, Y. Zhang, K.J. Lin, Efficient algorithms for web services selection with end-to-end QoS constraints, ACM Trans. Web 1 (1) (2007), Article 6, 26 pp.
[74] Z. Zakaria, S. Petrovic, Genetic algorithms for match-up rescheduling of the flexible manufacturing systems, Comput. Ind. Eng. 62 (2012) 670–686.
[75] K.J. Zhang, P.J. Dong, B. Ma, B.Y. Tang, H. Cai, Innovation of IT service in textile industrial clusters from the service system perspective, in: 2010 International Conference on Logistics Systems and Intelligent Management, 3, 2010, pp. 1819–1822.
[76] X.Y. Zhang, Z.W. Mo, F. Xiong, W. Cheng, Comparative study of variable precision rough set model and graded rough set model, Int. J. Approx. Reason. 53 (1) (2012) 104–116.
[77] X. Zhao, E. Liu, G.J. Clapworthy, M. Viceconti, D. Testi, SOA-based digital library services and composition in biomedical applications, Comput. Methods Progr. Biomed. 106 (3) (2012) 219–233.
[78] W.P. Ziarko, Rough Sets, Fuzzy Sets and Knowledge Discovery, Springer-Verlag, New York, 1999.
[79] P. Zou, Y. Gan, Y. Chen, B. Zhang, Dynamic composition of Web services using efficient planners in large-scale service repository, Knowl. Based Syst. 62 (2014) 98–112.

Please cite this article in press as: C.-S. Chen, et al., A cloud computing platform for ERP applications, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.11.009