Configuration as a Service in Multi-Tenant Enterprise Resource Planning System

2 authors:

Mona Misfer
King Saud University
1 PUBLICATION **7** CITATIONS

SEE PROFILE

Djamal Ziani
Al-Yamamah University
13 PUBLICATIONS **57** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project ERP Cloud Security View project

Configuration as a Service in Multi-Tenant Enterprise Resource Planning System

Mona Misfer Al-Shardan and Djamal Ziani

Abstract—Enterprise resource planning (ERP) systems are the organizations tickets to the global market. Thus, organizations can manage and coordinate all functions, processes, resources and data from different departments by a single software. However, many organizations consider the cost of traditional ERP is expensive and look for alternative affordable solutions within their budget. One of these alternative solutions is providing ERP over a software as a service (SaaS) model. A key feature of any SaaS system is the multi-tenancy architecture where multiple customers (tenants) share the system software. However, the SaaS developers accommodate each tenant's unique requirements by allowing tenant-level customization or configuration. While customization requires source code changes and in most cases a programming experience, the configuration process allows user to change many features within a pre-defined scope in an easy and controlled manner. Although literature provides many techniques to accomplish the configuration process in different SaaS systems, the nature and complexity of SaaS ERP are merely described in previous researches. Thus, this research is built on strong knowledge regarding the configuration in SaaS to define specifically the configuration borders in SaaS ERP and to design a configuration service with the consideration of the different configuration aspects.

Index Terms—Configuration, software as a service, multi-tenancy, ERP.

I. INTRODUCTION

Enterprise Resource Planning (ERP) is one of the largest IT investment that has been adopted by organizations for over a decade [1]. By integrating and controlling all the primary business processes in a single application, ERP facilitates the flow of information across the entire organization which enhances the organization's productivity and customer satisfaction and as a result the organization's position in the market [2]-[4]. Despite these advantages and many others, researches agreed on the fact that the implementation of traditional ERP system is considered expensive and time-consuming. Therefore, many ERP vendors are looking for alternatives to attract organizations (specially small and medium size enterprises) to invest in their ERP systems.

With the evolution of cloud computing, cloud ERP has a chance to become a mainstream affordable option.

Cloud computing is a new computing method for delivering a special services using internet media and data

storage on a third party server [3], [5]. The most popular delivery model of cloud computing is the Software as a service (SaaS). SaaS is defined by the national institute of standards and technology (NIST) as "a cloud computing service model that gives the consumer the capability to use the provider's applications running on a cloud infrastructure and accessible from various client devices through either a thin client interface (e.g. web browser) or a program interface" [6].

The main characteristic of SaaS model that gives it the cost advantages is the multi-tenant architecture while the same software can be running by multiple customers [7], [8]. However, one instance of a software may not meet every company different business needs and requirements. Thus, each company needs to tailor its system to meet their requirements and to differentiate it from other competitors [7]. This tailoring approach can leverage in two majors methods: configuration and customization. Although these two terms are frequently used interchangeably, different vendors may use different explanation for each term [9]. However, it is important to understand the configuration aspects in SaaS ERP for both ERP SaaS vendors and customers in order to manage their systems efficiently and deliver full value to their businesses. According to [9], the customization approach requires source code modification, while configuration is applied through a pre-defined parameters setting to change application functions within pre-defined scope [9]. While customization of SaaS ERP is one of the main problems that organizations have been complaining about due to its cost and complexity [3], [7], [10], the configuration process is considered as one of the key success of any SaaS ERP [11]. Some studies have addressed different configuration methods in different SaaS systems. However, studies didn't address the details of configuration methods for SaaS ERP with the concern of the multi-tenancy environment. While the ERP has multiple services integrated in one software, the configuration issues regarding SaaS ERP are different than other SaaS systems and should have more concerns. Regarding that, this research address two questions:

Q1: What is meant by configuration in SaaS ERP?

Q2: how to design a configuration service in SaaS ERP?

In the following sections, the paper starts with a background summarizing the basic aspects in the cloud computing and ERP context in Section II. In Section III the nature of customization and configuration in ERP SaaS based on its architecture is presented. The previous related works are shown in Section IV. Finally, the research proposed architecture design to support configuration in ERP SaaS in multi-tenant environment in Section V is shown and discussed.

II. BACKGROUND

This section will discuss the main two aspects regarding this research: A) cloud computing with concerned on the SaaS model and some related terms, and B) Enterprise resource planning system.

A. Cloud Computing

The National Institute of Standards and Technology (NIST) defined cloud computing as a "*model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction*" [6]. Most researches categorize cloud computing into three classes (based on the deployment models): Public cloud, Private cloud and Hybrid cloud [5], [8], [12].

There are multiple services provided through cloud computing. These services are collected into three main services: Infrastructure as a service (IaaS), Platform as a service (PaaS) and Software as a service (SaaS). While this study is exploring ERP system in SaaS model, the following is a description of this model and some key technologies regarding it.

B. SaaS Definition

Software as a service is about providing an application running on a cloud infrastructure. Thus, the cloud consumer is able to reach the application usually via a browser. With SaaS, there is no need to purchase the application license or install the application on the consumer's own computers and these application are usually shared by multiple clients using the (multi-tenancy architecture) [6], [8], [13], [14].

C. Multi-Tenancy Architecture

(Bezemer & Zaidman) defined the multi-tenant application by its main feature which is "*the application that allows customers (tenants) to share the same hardware resources, by offering one shared application and database instance*" [15]. In other words, in multitenant environment different organizations requests are served by one application instance resides on a shared software and hardware infrastructure which is isolated from other shares with regard to performance and data privacy [16]-[18]. Nevertheless, SaaS application with multi-tenant emphasizes on the idea of the (pay as you use). Customers can purchase needed modules and different customers use different function modules set [19].

D. Roles in SaaS Model

In SaaS model, there are three roles involved [20]:
1) The SaaS application vendor which is the company that developed the SaaS application.
2) The SaaS provider which is the company that runs and sells the software as a service.
   Note: the first and second roles can be the same company.
3) The SaaS customer/tenant which is the person or the company that subscribes and uses the SaaS application.

E. Configuration Objects in SaaS

Configuration objects in SaaS can be studied under three different aspects which are user interface, work flow and access control as explained in detail next.

1) User Interface (UI): Configuring UI refers to changing its look and feel [11], [21], [22]. Typically this include: pages layout, theme, titles, fonts, add/delete hyperlinks, etc [11], [21]-[23]. Moreover, UI configuration can be either for adding/deleting/editing components on the forms or adding/deleting/editing forms in the UI [11], [21]. In SaaS ERP systems, UI configuration can be embedded into the front-end configuration [17]. However, the UI configuration could be considered complex in SaaS systems, hence each tenant vary in their UI requirements [24].

2) Workflow: A workflow includes a set of activities, business objects, roles and rules that are unique to the organization. Changing the workflow may change the application behaviour [11], [22].

3) Access control or (organization structure): This includes both the account type and the allowed accessible resources for it. Thus, for each tenant there is specific access control data and each tenant should be able to create/edit/delete roles according to the organization structure [11], [21].

F. Enterprise Resource Planning (ERP)

ERP is one of the largest IT investment that is adopted by organizations for over a decade [1]. EPR defined by (Klaus, Rosemann, and Gable) as "*comprehensive, packaged software solutions seek to integrate the complete range of a business's processes and functions in order to present a holistic view of the business from a single information and IT architecture*" [25]. It facilitates the flow of information across the entire organization by integrating and controlling primary business processes in a single application and a single data repository [3], [4]. However, many researches agreed on the fact that the cost of ERP system is considered expensive. This cost starts from buying the system, installing it, considering the hardware cost and the consultants fees and it doesn't end in this point, the organization must consider also the need for IT staff to operate, maintain, upgrade and debug the system. The panorama report of 2013 claims that the majority of ERP projects (53%) exceed their project budgets [26]-[28]. However, cloud computing is becoming a dominant choice of deployment for ERP software of most ERP vendors such as SAP, Oracle, Microsoft,...etc. Moreover, a recent study completed by Gartner advises to "*consider cloud ERP as a potential replacement for on-premise ERP systems that are out of support or running on an old technology platforms (such as mainframes)*" [29].

Fig. 1. Examples of ERP modules.

An ideal ERP system covers different functional areas

(ERP modules Fig. 1). However, different ERP vendor may provide different names for the same module or different integrated modules [30]. In typical ERP system, the various modules are integrated across an enterprise through a single database [30], [31].

III. CONFIGURATION AND CUSTOMIZATION

Tailoring or adjusting SaaS application to match the customer specific needs is a general approach which can be done using customization or configuration or both [9], [32].

Customization and Configuration are two terms that are frequently used interchangeably. Different vendors may use different explanation for each term [9]. In the literature it has been found that customizations is usually described as the process of implementing a new feature to the application that doesn't even exist which requires changes in the source code with deep understanding of the exciting program functionality and the domain the program should support [9], [32], [33]. This ability is limited to the application vendor most of the times hence the changes are made in the second phase of the application lifecycle (implementation phase) [32]. On the other hand, configuration allows the user to adjust the application through pre-defined parameters to change the application functions within pre-defined scope and it doesn't require a source code changes [9]. However, some researches point at the customization as a general term for adjusting a system and the configuration is just one of the customization methods [34], [35]. Ref. [34] defined customization in general as "*Adapting standard software to the requirements of an individual organization*" and point to the ability of configuring the software in alternative name (parameterization) which means the settings of parameters or selecting from list of options [34]. In similarity define the configuration as a process that has two key features:

1) Assembling the object from instances of fixed, well defined component types.
2) These components interact with each other in predefined way.

Based on these features, the configuration process is about providing a list of selected components [36]. In the SaaS context, configuration is aimed to provide each tenant with multitude of options and variations without changing the SaaS application source code and this modification should done by the tenant at run-time without suspend the application during the configuration [11], [21], [37]. With the multi-tenant architecture, some researchers found that the customization must be made possible only through the configuration [15], [38].

Previous attempts found in the literature to distinguish the differences between customization and configuration is the work of [9], who claims that the difference between the customization and configuration is all about the level of complexity of application changes [9]. However, in general there is a lack in the literature on the topics that consider customization and configuration in cloud ERP [32].

For the reason of this study we defined the configuration in SaaS ERP based on three aspects (Fig. 2): 1) What are the elements of configuration, 2) How to configure these elements and 3) Who will be allowed to configure.

IV. RELATED WORK

Although there are many studies about the configuration and customization in different SaaS systems, not many researches entered the field of configuration of ERP over SaaS and particularly in the multi-tenancy environment. However, [9] provided a generic framework to guide SaaS vendors in planning and executing configuration and customization strategies. In [9], the authors abstracted the potential approaches around configuration and customization into four models. Three of them involve changing the application source code by the SaaS vendor according to each tenants' requirements or according to a big group of tenants' requirements. The fourth suggested approach (which is the concern of this research) involves providing web based tools and API for tenants to conduct configuration and customization by themselves [9]. Reference [24] also proposed a guided configurable solutions in SaaS application. The authors in [24] provided a combination of configurable solution based on XML for UI and functions with a configurable solution based on metadata for data configuration. They applied their proposed solution on a graduation project management system [24]. However, the details of the implementation are merely mentioned. Also, [21] provided a general information about the nature of configuration in Electronic Contract Management application over SaaS software. They described the different configuration aspects for SaaS application (UI, workflow, data and access control) and provided a metadata customization module for each of these aspects in Electronic Contract Management application [21]. [11] in similarity classified different configurable aspects of SaaS applications and proposed SaaS application architecture to support configurability by the tenant's designer. They applied their proposed architecture on a university grading management system. In their proposed architecture, their configuration approach depends on a set of predefined templates. So, the tenant's designer can use these templates in the edit mode of the application and by the designer UI to configure the application, while the end user uses the configured data in the play mode of the application [11]. However, their approach didn't specifically discuss the multi-tenancy in such SaaS applications.

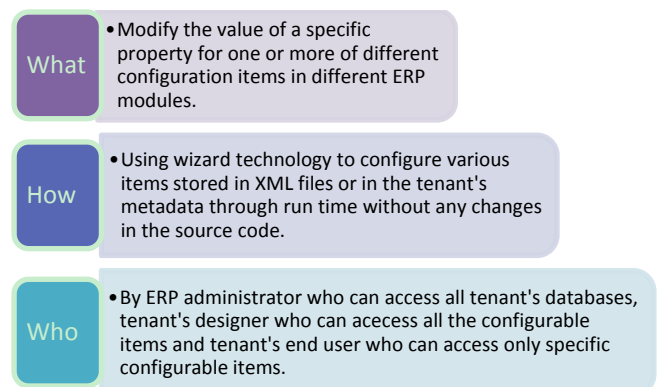| What	• Modify the value of a specific property for one or more of different configuration items in different ERP modules.
How	• Using wizard technology to configure various items stored in XML files or in the tenant's metadata through run time without any changes in the source code.
Who	• By ERP administrator who can access all tenant's databases, tenant's designer who can acecess all the configurable items and tenant's end user who can access only specific configurable items.

Fig. 2. Research definition of configuration in SaaS ERP.

In contrast, [39], [40] consider the configuration within multi-tenant applications and they both used similar methods while they focused on configuration in three aspects : data, function and user interface. Regarding the data configuration, they used the expansion table method which allows different

tenants to store different data within customized fields. However, [40] didn't go through details of how to implement the expansion table, while [39] applied this method on an online service for human resource. Their ideas depend on the metadata configuration to configure the user interface. However, although [39] showed an example of the case study it didn't provide a solution for interdependent functions while [40] provides only a generic description of relations between functions. In similar work, [19] focused on the configuration of functions and user interface in SaaS application. However, the details in how to configure the specific elements in UI were omitted. On the other hand, a conceptual multi-tenant SaaS architecture application platform presented by [22] to enable the execution of configurable and multi-tenant SaaS application. Their proposed platform allows configuration in five aspects in SaaS applications. Four of these aspects are similar to [11], [21] which are : UI, workflow, data and access control which they named (organizational structure). Moreover, [22] considered the configuration in the business logic which is the functions that handle information exchange between the database and UI [22]. Although the paper merely describes the details of how to deal with the workflow configuration and UI configuration, the authors provide an example scenario of the SaaS platform interaction described all phases of the application lifecycle based on the metadata driven architecture presented in force.com [22]. Moreover, [41] proposed a high level of SaaS maturity model that supposed to support configurability using metadata and Java Management Extensions (JMX) to manage the metadata. They applied the model on a Supply Business Management system (SBM) [41].

V. PROPOSED ARCHITECTURE DESIGN

This research proposed architecture is based on a platform that supports multiple databases for each tenant. Thus, each tenant will use a separate database to save the transaction and the configuration data. The tenant can see only its own configuration, which is saved in a secure database with restricted access.
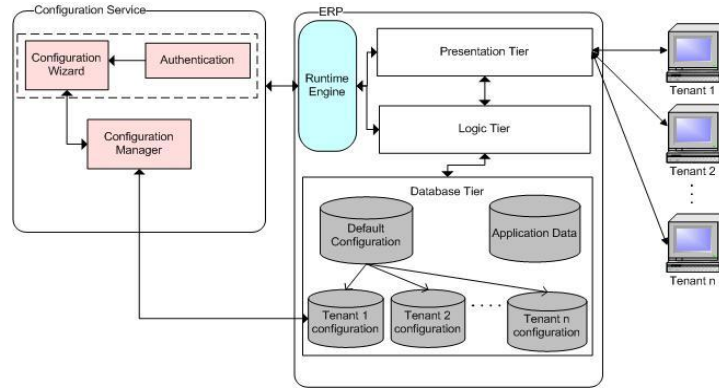

Fig. 3. High level architecture of the configuration service in SaaS ERP.

However, these databases are connected with the configuration service which is composed of three main modules (authentication, configuration wizard and configuration manager). Fig. 3 shows a high level architecture of the proposed solution and detailed descriptions of the architecture are given in the following subsections.
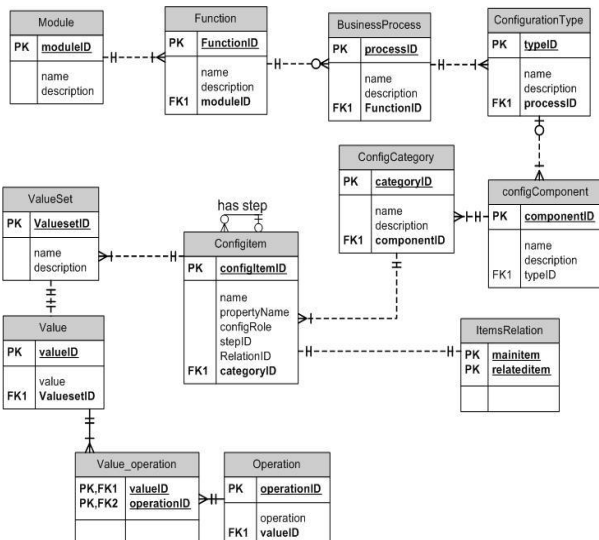

Fig. 4. Configuration model.

A. Authentication

To avoid complexity and develop a central configuration management, there is a need to restrict the number of options and features that could be configured. Thus, three types of users identified and each user type has different accessibility to the configuration items based on the attribute (configureRole) assigned to each item. The user types are: 1) ERP administrator: is the person who has the privilege to manage all the items and assign the configurable items' role to the other types. 2) Tenant's designer who can access all the configurable items and 3) Tenant's end user who is only have access to some configurable items (often the user interface items). The tenant identification will check and pass to the configuration wizard by the authentication module.

B. Configuration Wizard

The configuration wizard is the core module of the configuration service. A wizard is a user interface that leads the user through a series of well-defined steps. Our proposed configuration wizard is built on a model that guides the user from the selected ERP module to the attribute of the configuration item to change its value. Fig. 4 shows the suggestion model. To categorize the model we suppose that each ERP module (e.g. supply chain management, human resources, customer relationship management) has multiple functions (e.g. in supply chain management there are:

purchasing, inventory, supplier scheduling,… etc) and for each function there are one or more business process (e.g. for purchasing function there are: negotiating price, billing process… etc). When the user reaches the selected business process which will be configured, s/he can select the configuration type (workflow configuration, access control configuration or user interface configuration). For each configuration type the user should specify the configuration component (e.g. table, page,.. etc).

Also, there are a set of pre-defined configuration categories for the configuration component (e.g. hide/show field, change button label,…etc) so the user can select the configuration category and specify the item related to this category. To be more specific, the user will not only select the configuration item but also specify the property that s/he wants to configure in that item. The user will have the opportunity to change the property's single value or multiple values (e.g. drop down list). Also, some configuration items may need more than one step to be configured (e.g. changing the values of drop down list). Thus, the user will navigate through all the configuration steps. Finally, the configuration process will not be completed if the configured item has a relationship (i.e. depend-on) with other configuration item. If so, the related item will be displayed to be configured.

C. Configuration Manager

After collecting the configuration data from the user, the configuration manager is responsible for storing and retrieving the configuration data to/from the tenant specific configuration database. Also, configuration manager controls the access to each tenant database so the tenant will be able to configure its own data in its own database without influencing other tenant's data.

D. Runtime Engine

The runtime engine plays a role in generating an instance of the application to the requester tenant using the tenant's configuration data in contrast with the application data.

E. An Example Scenario

The following is present the configuration process from the beginning and how the user can perform the changes running through different steps.

The configuration scope is pre-codify by the ERP developer in design time according to the proposed configuration model and store this configurable items in a separate database as default configuration. When a customer registers to the ERP services, a copy of the default configuration stored in a tenant specific database.

The configuration process starts with checking the user type by the authentication module as described earlier. Then, the user can navigate through a certain sequence of allowable configuration categories until a specific configuration item is selected. The user can only make changes in two parameters of the selected item: 1) the value of a specific property regarding the configurable item and 2) the operation related to that value (e.g. >, <, = … etc). The guided wizard and predefined constraints will ensure the consistency, quality and easiness of the configuration process. Thus, the ERP administrator can manage this process and add configurable items and constraints as much as needed. The ERP administrator also can manage the access control by adding access roles to each configurable item. Furthermore, when the user wants to retreat the configuration, the configuration manager reloads the default configuration of the configurable item into the tenant configuration database.

For clarifying this, suppose a tenant designer wants to hide the field "external order" in the order page .The following is a hierarchical view of configuration levels.

ERP Module: Supply chain management
    Module Function: Purchasing
    Business Process: Create order
    Configuration Type : User Interface
    Configuration component: create order page
    Configuration Category: Show/Hide fields
    Configuration item:"External Order" field
    property name: visibility
    value: hidden
    operation: equal

VI. CONCLUSION AND FUTURE WORK

This paper explored the configuration of the enterprise resources planning systems in the context of SaaS multi-tenancy environment. A configuration service architecture is proposed based on specific criteria learned from the literature. We adopted the wizard technology to be the portal that interprets the proposed configuration model. Based on that model the configuration is within a scope defined through pre-codify configurable items by the ERP developers during building and deploying the services. The performance issues have been treated by organizing the configurable aspects in a hierarchal way. Thus, separating each configuration category and enabling the user to change only two parameters of the selected item at a time. This process will speed up the load of application changes. Another way to enhance the performance is tenant database isolation technique which will guarantee the security and privacy for each tenant's configuration data as well. Moreover, dealing with isolated small configuration files will surly speed up storing and retrieving these files.

For future work, we plan to address the validation issues and adding a mechanism that ensures the quality of configuration made by the user and the valid of configuration in logic and consistency.

REFERENCES

[1] M. Al-Mashari, A. Al-Mudimigh, and M. Zairi, "Enterprise resource planning: A taxonomy of critical factors," *Eur. J. Oper. Res.*, vol. 146, no. 2, pp. 352-364, Apr. 2003.
[2] K. Metaxiotis, "Introducing ERPs to SMEs: A two-dimensional review and analysis," *Bus. Rev. Cambridge*, vol. 17, pp. 83-90, 2011.
[3] G. Purohit, M. Jaiswal, and M. Pandey, "Challenges involved in implementation of ERP on demand solution: Cloud computing," *Int. J. Comput. Sci.*, vol. 9, no. 4, pp. 481-490, 2012.
[4] R. Addo-Tenkorang and P. Helo, "Enterprise Resource Planning (ERP): A review literature report," in *Proc. World Congr. Eng. Comput. Sci.*, vol. 2, 2011.
[5] E. F. Kiadehi and S. Mohammadi, "Cloud ERP: Implementation of enterprise resource planning using cloud computing technology," vol. 2, no. 11, pp. 11422-11427, 2012.
[6] P. Mell and T. Grance, "The NIST definition of cloud computing," *NIST Spec. Publ.*, 2011.
[7] M. Gd and M. Bist, "EAAS-ERP as A Service," *J. Inf. Oper. Manag.*, vol. 3, no. 1, pp. 141-145, 2012.
[8] G. F. H. Raihana, "Cloud erp – A solution model," *J. Comput. Sci.*, vol. 2, no. 1, pp. 76-79, 2012.

[9] W. Sun, X. Zhang, C. J. Guo, P. Sun, and H. Su, "Software as a service: Configuration and customization perspectives," in *Proc. 2008 IEEE Congr. Serv. Part II (services-2 2008)*, Sep. 2008, pp. 18-25.

[10] A. Lenart, "ERP in the cloud – Benefits and challenges," *Research in Systems Analysis and Design: Models and Methods*, 2011, pp. 39-50.

[11] Nitu, "Configurability in SaaS (software as a service) applications," in *Proc. The 2nd India Software Engineering Conference (ISEC '09)*, 2009, pp. 19-26.

[12] B. Rimal and E. Choi, "A conceptual approach for taxonomical spectrum of cloud computing," in *Proc. 4th International Conference on Ubiquitous Information Technologies & Applications ICUT '09*, Fukuoka, Japan, 2009.

[13] T. Dillon, C. Wu, and E. Chang, "Cloud computing: Issues and challenges," in *Proc. 2010 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, 2010, pp. 27-33.

[14] W.-T. Tsai, X. Sun, and J. Balasooriya, "Service-Oriented cloud computing architecture," in *Proc. 2010 Seventh Int. Conf. Inf. Technol. New Gener.*, 2010, pp. 684-689.

[15] A. Bezemer and C. P. Zaidman, "Multi-tenant SaaS applications: Maintenance dream or nightmare?," in *Proc. the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE)*, 2010, pp. 88-92.

[16] R. Mietzner, A. Metzger, F. Leymann, and K. Pohl, "Variability modeling to support customization and deployment of multi-tenant-aware software as a service applications," *Princ. Eng. Serv. Oriented Syst.*, vol. 215483, pp. 18-25, 2009.

[17] J. Müller, J. Krüger, and S. Enderlein, "Customizing enterprise software as a service applications: Back-end extension in a multi-tenancy environment," in *Proc. International Conference on Enterprise Information Systems - ICEIS*, 2009, pp. 66-77.

[18] R. Krebs, C. Momm, and S. Kounev, "Architectural concerns in multi-tenant SaaS applications," *CLOSER*, 2012.

[19] H. Wang and Z. Zheng, "Software architecture driven configurability of multi-tenant SaaS application," pp. 418-424, 2010.

[20] R. Mietzner, F. Leymann, and M. P. Papazoglou, "Defining composite configurable SaaS application packages using SCA, variability descriptors and multi-tenancy patterns," in *Proc. 2008 Third Int. Conf. Internet Web Appl. Serv.*, Jun. 2008, pp. 156-161.

[21] P. Arya, V. Venkatesakumar, and S. Palaniswami, "Configurability in SaaS for an electronic contract management application," in *Proc. 12th Recent Advances in Networking, Vlsi and Signal Processing*, 2010, pp. 210-216.

[22] S. Kang, S. Kang, and S. Hur, "A design of the conceptual architecture for a multitenant saas application platform," in *Proc. 2011 First ACIS/JNU Int. Conf. Comput. Networks, Syst. Ind. Eng.*, May 2011, pp. 462-467.

[23] W. -T. Tsai and X. Sun, "SaaS multi-tenant application customization," in *Proc. 2013 IEEE Seventh Int. Symp. Serv. Syst. Eng.*, Mar. 2013, pp. 1-12.

[24] J. K. Zhou, G. X. Qiu, Z. T. Hu, and B. Zhou, "Research and implementation of configurable technology based on SaaS," *Appl. Mech. Mater.*, vol. 198-199, pp. 500-505, Sep. 2012.

[25] H. Klaus, M. Rosemann, and G. Gable, "What is ERP?" *Inf. Syst. Front.*, 2000.

[26] S. Trimi, S. M. Lee, D. L. Olson, and J. Erickson, "Alternative means to implement ERP: Internal and ASP," *Ind. Manag. Data Syst.*, vol. 105, no. 2, pp. 184-192, 2005.

[27] D. Aloini, R. Dulmin, and V. Mininno, "Risk management in ERP project introduction: Review of the literature," *Inf. Manag.*, vol. 44, no. 6, pp. 547-567, Sep. 2007.

[28] D. Danaiata and C. Hurbean, "SaaS – Better solution for small and medium-sized enterprises," *Applied Economics, Business and Development*, 2010, pp. 29-34.

[29] N. Rayner. Survey analysis: Adoption of cloud ERP, 2013 Through 2023. [Online]. Available: https://www.gartner.com/doc/2656317.

[30] S. Kumar, S. Nirmalkar, and S. K. Meesala, "Enterprise resource planning- A complete business solution," vol. 2, no. 3, pp. 406-416, 2013.

[31] G. Juell-Skielse and H. Enquist, "Implications of ERP as service," *Re-conceptualizing Enterp. Inf. Systems*, 2012, pp. 129-151.

[32] M. Mijač, R. Picek, and Z. Stapić, "Cloud ERP system customization challenges," in *Proc. Uropean Conference on Information and Intelligent systems*, 2013, pp. 132-140.

[33] Y. Dittrich, S. Vaucouleur, and S. Giff, "ERP customization as software engineering: Knowledge sharing and cooperation," *IEEE Software*, vol. 26, no. 6, pp. 41-47, 2009.

[34] K. E. Kurbel, The Making of Information Systems Software Engineering and Management in a Globalized World, Springer, 2008.

[35] W.-T. Tsai and X. Sun, "SaaS multi-tenant application customization," in *Proc. 2013 IEEE Seventh Int. Symp. Serv. Syst. Eng.*, Mar. 2013, pp. 1-12.

[36] D. Sabin and R. Weigel, "Product configuration frameworks-a survey," *IEEE Intell. Syst.*, pp. 42-49, 1998.

[37] X. Zhang, P. Sun, Y. Huang, and W. Sun, "A model-driven framework for enabling self-service configuration of business services," in *Proc. 2008 IEEE Int. Conf. Web Serv.*, Sep. 2008, pp. 497-504.

[38] C. -P. Bezemer, A. Zaidman, B. Platzbeecker, T. Hurkmans, and A. T. Hart, "Enabling multi-tenancy: An industrial experience report," in *Proc. of2010 IEEE Int. Conf. Softw. Maint.*, Sep. 2010, pp. 1-8.

[39] S. Zhang, Z. Li, and X. Chen, "Research of configuration on multi-tenant," *Inf. Technol. J.*, vol. 10, pp. 2154-2160, 2011.

[40] S. W. Zhang and X. P. Wang, "Configuration of multi-tenant applications," *Adv. Mater. Res.*, vol. 219-220, pp. 1182-1185, May 2011.

[41] Y. Zhang, S. Liu, and X. Meng, "Towards high level SaaS maturity model: Methods and case study," in *Proc. 2009 IEEE Asia-Pacific Serv. Comput. Conf.*, Dec. 2009, pp. 273-278.

Mona Misfer Al-Shardan is a teaching assistant in the Department of Information Technology, College of Computer and Information Sciences, King Saud University since 2011. She obtained her bachelor's degree from the Computer Sciences Department, Princess Nora Bint Abdulrahman University in 2008. Currently she is a master's student in the Department of Information Systems, College of computer and Information Sciences, King Saud University. Her current research is centered on configuration in cloud computing systems especially enterprise resources planning systems.

Djamal Ziani is an assistant professor in the Department of Information Systems, college of Computer and Information Sciences, King Saud University since 2009. He is a researcher in data management group of CCIS King Saud University.

He received master degree in computer sciences from University of Valenciennes France in 1992, and Ph.D. in computer sciences from University of Paris Dauphine, France in 1996.