

ARCHITECTURAL DESIGN OF EXPERIENCE BASED FACTORY MODEL FOR SOFTWARE DEVELOPMENT PROCESS IN CLOUD COMPUTING: INTEGRATION WITH WORKFLOW AND MULTI-AGENT SYSTEM

M. Hanafiah, R. Abdullah*, J. Din and M. A. A. Murad

Faculty of Computer Science and Information Technology, University Putra Malaysia, 43400 Serdang, Selangor, Malaysia

Published online: 05 October 2017

ABSTRACT

A model which is based on experience factory approach has been proposed earlier, called EBF-SD, to overcome the limitations of experience management in software development domain. An application prototype, which is then called SDeX, is developed based on the proposed model. The study on correlation analysis indicates that automation do have positive relationship with other components: knowledge management, cloud, collaboration and portal. This paper further discusses the high level prototype development with the emphasis on the architectural design. Automation features are incorporated in the design in which workflow system and intelligent agents are integrated, and the facilitation of cloud environment is empowered to further support the automation.

Keywords: architectural design; knowledge management; experience factory; workflow; multi-agent system; cloud automation.

Author Correspondence, e-mail: rusli@upm.edu.my

doi: <http://dx.doi.org/10.4314/jfas.v9i4s.12>

1. INTRODUCTION

Knowledge management (KM) for software development (SD) has emerged since the late 1990s and enormous researches have been established since then. Due to the nature of SD process in which knowledge intensive activities are involved, there exists countless issues in knowledge transfer and sharing. Challenges in KM occur even more frequent for distributed



teams as summarized in [1] in which the issues are related to knowledge transfer and information flow, explicit and tacit KM, inconsistencies and misinterpretation. As pointed in [1], these problems occurs due to the differences in cultures [2-3], project complexity [4] and inefficient communication [5-6]. Many collaborative solutions have been introduced so far: e.g. domain-specific ontology for distributed development [2], conceptualized flow models [5], integrated enterprise software engineering models and effective quality measures [7], and multi-agent context processing mechanisms [8]. These solutions however offer generally on “knowledge pull” rather than “knowledge push” in which knowledge awareness and continuous learning are not fully emphasized. In view of this limitation, a model has been proposed to facilitate knowledge management for SD process with the right and appropriate technology empowered with automation features to support some of the human tasks.

KM is mainly driven by voluntarily actions and it is hardly committed by the employees without adequate support and motivation. As part of KM initiatives, KM process which facilitates the creation, storage, retrieval and reuse of experiences and knowledge can be automated in a way that human intervention is minimized so that awareness and higher commitment can be expected from the employees. Automation is defined as "the creation and application of technology to monitor and control the production and delivery of products and services" [9]. Automation in KM system (KMS) can be achieved via the integration of workflow management and agent technology. To further enhance the automation implementation, cloud computing environment could be leveraged for a more efficient and effective automation approach.

In KM, there are three important components: agents, business processes and knowledge assets. Knowledge assets are possessed by agents, and the agents participates in business process where the knowledge is used and updated [10]. In [15] emphasizes that for workers to make full use of KM for the purpose of providing better judgment and decisions, Workflow Management System (WfMS) can be viewed as a trigger of KM-support activities and a distributor of organizational knowledge. Workflow is a series of activities required to complete a task. In an organization, workflow is a concept of defining, executing and automating the business process where tasks, information or documents are passed from one participant to another for action with a set of rules or guidance to complete them. WfMS coordinates the business process by defining, managing, executing structured workflows and ensures that the right tasks are executed at the right time by the right people [10]. Workflow management benefits both the software applications and overall business goals: it eases the

costs developing new enterprise applications and reduces the maintenance of standard enterprise software [11].

However, according to [12], the workflow approach as business process automation merely describes the process from a centralized perspective, in a way that a complete list of activities and paths are provided, the criteria to follow the path are defined and the ordering constraints are given. In such a case, it is argued that workflow solution seems lack of reactivity, semantics, extensibility, resource management and heterogeneity. In order to make the components more active, it has been proposed that the integration of autonomous agents into the workflow system is necessary.

Intelligent agents have been used in KM solutions to manage the complexity of the system. Several works on multi-agent systems (MAS) have been done previously e.g. in [7, 16-17]. Agents can be viewed as autonomous, problem-solving computational entities equipped with social abilities to offer effective proactive behaviour in open and dynamic environments [18]. Autonomous agents exhibit the following properties below to represent their role and responsibility [13]:

- **Autonomy:** Most problem-solving tasks are performed without intervention from humans or other agents; they have control on their own state.
- **Social ability:** Agents interact when appropriate with other agents to help solving their problem or to help other.
- **Pro-activeness:** Agents take initiative and exploit unexpected opportunities where appropriate.
- **Responsiveness:** Agents perceive their environment and respond in a timely fashion to changes in it.

A multi-agent system (MAS) is a loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each software agent [17]. These agents usually have a small knowledge base with a specified intelligence that collaborates with other agents to ensure the consistent and coherent knowledge based, also it facilitates the communication and coordination between the agents. In MAS, agents interact among each other to achieve their objectives through cooperation, negotiation and coordination [19]. Agent technology is necessary to reduce costs, to improve efficiency and effectiveness, to support the requirement collaboration globally and to support IT applications that are previously could not be developed [20]. Integration between workflow and agent has been developed previously in KM based applications as in [10, 20]. The incorporation

between workflow and agent technology ensures the automation objectives are met and that the integration between people and business process applications are more streamlined.

Technology today has moved from in-premise operations to cloud computing environment. Cloud computing is computing in which large groups of remote servers that are networked to allow centralized data storage and online access to computer services or resources [21]. Cloud computing has the following elements [22]: on-demand self-service, broad network access, resource pooling, rapid, elasticity and measured services.

The three main levels of cloud services include: (i) Application/Software as a Service (SaaS) such as Google Apps, Amazon Web Services and Salesforce.com, (ii) Platform as a Service (PaaS) such as Google App Engine, Microsoft Azure Service and (iii) Infrastructure as a Service (IaaS) such as AWS Elastic Beanstalk and Windows Azure [23]. These cloud elements and services will be able to handle the required automation for KM applications. Example scenarios on cloud automation can be observed from two scenarios below [24]:

- 1) Customer-centric processes that can deliver multichannel experiences: business process can be easily accessible from multiple devices by utilizing SOAP APIs without having to worry about the implementation details.
- 2) Custom applications which include extension of SaaS and on-premise application: applications span a wide range of options, from simple approval forms to more complex workflows that integrate data from SaaS and on-premise applications.

Automation and cloud computing may go hand in hand to achieve solution that could provide technology designed with high performance, availability, greater reliability, rapid mass scalability and at a reduced cost that aids in higher growth and flexibility [25]. Cloud environment makes it possible for reliability and availability of the services without having to take care of the infrastructure. This would benefit the implementation of KMS automation features. Additionally, the contribution from [14] on cloud data security based on MAS architecture can be considered as value added features to facilitate confidentiality, correctness, availability and integrity of cloud data storage.

Aforementioned, an experience based factory model for SD process named as EBF-SD has been proposed in [1]. In this paper, further details on the prototype and architectural design of the model with the integration of automation process with workflow, intelligent agents and cloud computing are discussed.

2. METHODOLOGY

The research undergoes the steps as depicted in Fig. 1. Literature review focuses on automation in KM which has been discussed in the introduction section. Prior to this study, a conceptual design has been developed and the verification has been conducted in [26]. In this study, we shall briefly look into the correlation analysis among the components in focus using the data collected in the pilot study.

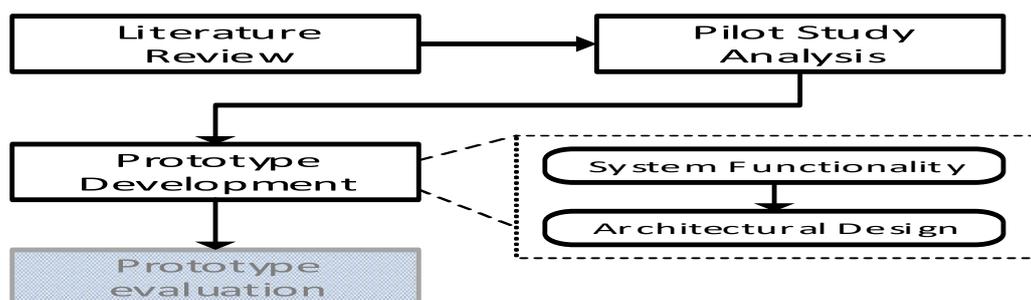


Fig.1. Research methodology

Correlation coefficients measure the strength of association between two variables. The Pearson product moment correlation coefficient value uses the common calculation as below. The correlation r between two variables is as shown below [32]:

$$r = \frac{\sum xy}{\sqrt{\sum x^2 \sum y^2}} \quad (1)$$

where $x = x_i - \bar{x}$, x_i is the x value for observation i , \bar{x} is the mean x value, $y = y_i - \bar{y}$, y_i is the y value for observation i and \bar{y} is the mean y value. When Pearson's r is close to 1, there is a strong relationship between the two variables. And when Pearson's r is close to 0, there is a weak relationship between the two variables. Null hypothesis in this study is 'Automation does not have significant relationship with the other component/sub-components: KM, Portal, Collaboration and Cloud'.

The research is further resumed with the development of the prototype. In this study, the prototype development focuses on the system functionality and the architectural design. The architectural design deals with high level architectural of the whole system, as well as emphasizing on how the automation elements are incorporated. Evaluation on the prototype will be done in future and is not the focus of this paper.

3. FURTHER PILOT STUDY ANALYSIS

In [26] has conducted a pilot study to verify the initial proposed model. Proposed components include Software Development Process (SD), Knowledge Management (KM), Experience Factory (EF), Community of Practise (CoP), Technology and Infrastructure (TI),

Organizational and Managerial (OM) influences. Further in this study, a correlation analysis has been conducted between Automation and KM as well as other sub-components: Portal, Collaboration and Cloud Computing.

Table 1 shows that the Pearson's r value between Automation and KM, Portal, Collaboration and Cloud is between 0.4 to 0.5 which indicating moderate relationship. The Pearson's r value between Cloud and KM, Portal, Automation and Collaboration is between 0.5 to 0.6 also indicating a moderate relationship. The correlations shown are not strong but sufficient enough to indicate positive relationship among the sub-components. The moderate correlations could be due to small sample size used (27 participants) in the pilot study. It is important to note that Sig (2-tailed) shows that the sub-component automation shows statistically significant (less than 0.01) with KM and Cloud, while it is slightly statistically less significant (less than 0.05) with Collaboration and Portal.

Table 1. Correlation analysis among sub-components

	KM	Portal	Automation	Collaboration	Cloud
KM	1				
Portal	0.488**	1			
Automation	0.522**	0.479*	1		
Collaboration	0.355	0.358	0.441*	1	
Cloud	0.555**	0.669**	0.547**	0.578**	1

**Correlation is significant at the 0.01 level (2-tailed).

*Correlation is significant at the 0.05 level (2-tailed).

The result however is sufficient to reject the null hypothesis and there is adequate evidence that there exists relationship between Automation and the other sub-components. Intuitively, the Cloud sub-component is statistically significant (less than 0.01) with all the components in focus: KM, Portal, Automation and Collaboration. This result is satisfactory to further proceed with the prototype development which is discussed in the next section.

4. PROTOTYPE DEVELOPMENT

The initial model, EBF-SD as proposed in [1] is then named as SDeX (Software Development eXperience Factory) in this study. For simplicity, 'knowledge and experience' are referred to 'experience' in this paper.

The main actors of SDeX are identified as follows:

- 1) Project Owner: a person who creates an SD project and owns the project.

- 2) Project Contributor: a person who chooses to be knowledge/experience contributor.
- 3) Project Subscriber: a person who chooses to subscribe to any project that he/she interested in. He/she will receive notification on any project update.

System functionalities have been divided into four main modules. The features are discussed below:

1) User and Project Setup:

- User ID is required for authentication.
- User authorization is required for access control.
- Project is created by project owner.
- User can be a project contributor or subscriber.
- Contributor or subscriber must obtain approval from project owner.
- User may self-manage themselves to indicate their interests and expertise in their profile setting.

2) Knowledge Base Taxonomy (Blueprint) Setup

- Master blueprint is prepared by a system administrator. For the initial development, two SD process models are considered: Scrum (agile) and Waterfall (traditional). Domain ontology on these process models becomes the initial knowledge base.
- Project owner should customize the project blueprint to suit his/her project needs.
- Finalized project blueprint is published for usage.

3) Experience Management Process

- Experience creation and dissemination.

Initial product experiences are created by project owner. Project contributors may deposit all experiences that they will be going through during the project lifecycle. Contributors may also rate the efficiency of the activities, techniques or methods. Completed product experiences are valued in terms of its efficiency or successfulness. Successful product experiences will be disseminated to the community subjected to the approval from the project owner. Approval, dissemination and notification are handled by a workflow and software agent.

- Experience search.

Search by keywords are possible including domain, technical or document knowledge search. Experts' correspondences are presented and instant interactions are possible via communication and collaboration tool (e.g. chat, emails, instant messaging, etc.)

- Experience reuse.

Completed product experiences are opened to the community for reuse. Reusable product experiences covers all the data, knowledge and experiences from the beginning of the product development until its end. When a product is reused, users are able to customize the product in their own project to suit their needs. Reuse can be triggered by users either by navigating through the experience base or by recommendation from software agent.

4) Statistics Update

This feature generates some of the relevant statistical reporting based on the software development process knowledge that are deposited into the system. Some examples of reports are: effort spent for every phase, efficiency of techniques and efficiency and productivity of sprints (for Scrum projects). This allows the organization to analyze the current practices, react appropriately and feedback accordingly to the relevant project teams.

5. ARCHITECTURAL DESIGN

5.1. Multi-Tier Architecture in Cloud Environment

The typical representation for enterprise-level application architectural is the layered multi-tier approach. As shown in Fig. 2, functionality is separated into 3 layers: presentation layer, application layer, and data layer. The multi-layer architecture has also been proposed by [33] in the research on security framework of cloud data storage which segregates the agent layer, data storage layer and cloud service provider. It is expected that modern UI applications to be as much flexible in a way that it can be accessed from web browsers as well as from mobile applications. Modern client technologies have shifted the Model-View-Controller (MVC) on the server to the MVC on the client [27].

Applications developed in cloud may expose the services via public API as endpoints to be called by different type of applications e.g. web browser, iOS or Android. In case of Google Cloud Platform, the service endpoints are exposed automatically in cloud endpoints via REST API. With the client MVC frameworks like AngularJS, the JavaScript code is used to integrate between the two technologies.

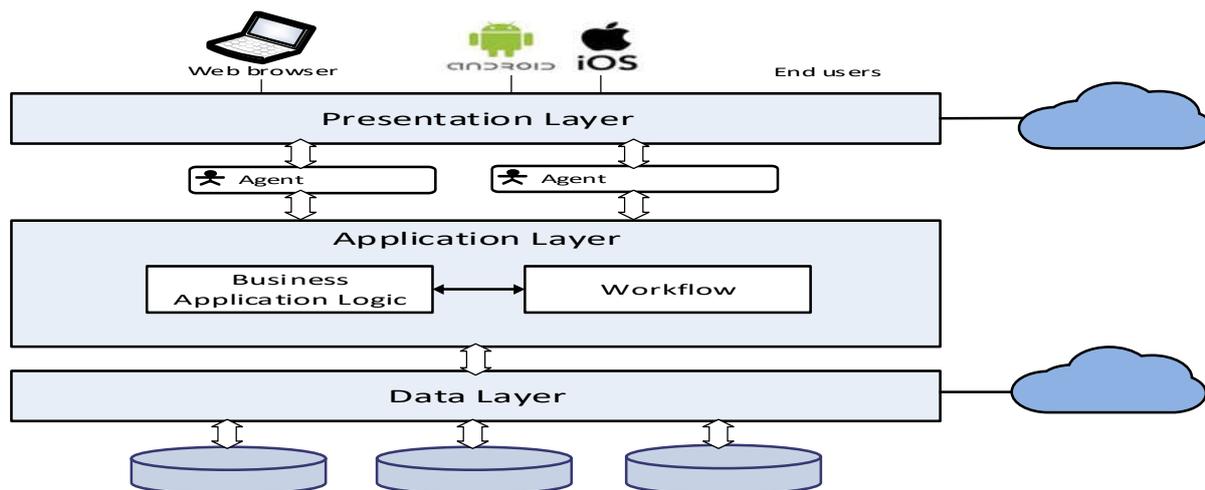


Fig.2. SDeX high level architecture

The application layer handles most of the business logic. Business services especially on the experience and knowledge management are supported by automation processes in a defined workflow and agent implementation.

Data layer deals with the storage of the application data. It also will make use of cloud services which can handle data storage efficiently. By using database services in the cloud, the application does not need to worry about the maintenance of the storage e.g. import/backup or failure recovery. Additionally, the use of cloud storage can support data that can grow to any size, provide high availability and enhance performance. The basic SD knowledge taxonomy, which is defined as ontology will be stored in the repository. As suggested by [34], it is recommended that building ontology can be done in stages as in capturing, coding and integrating with existing ontologies. In this case, the software engineering ontology developed by [35] is used as a reference.

Table 2 shows the summarized functionality versus the implementation approach.

Table 2. Functionality and implementation approach

Functionality	SDeX Service	Cloud Service	Work-Flow	Agent
User authentication		✓		
Project creation	✓	✓	✓	
Knowledge taxonomy	✓	✓		
	KM processes			
- search	✓			
- creation	✓	✓	✓	✓
- storage	✓	✓	✓	
- disseminate	✓		✓	✓

- reuse and apply	✓	✓	✓
Statistical/report updates	✓		✓

SDeX services are the public APIs that are exposed as endpoints. As described previously, these services can be called by multiple interfaces such as web, iOS or Android applications. Cloud service refers to the available services provided by the cloud. For example, user authentication can be realized with Google’s OAuth, an open standard for token-based authentication and authorization on the internet [28]. Other functionalities that will make use of cloud services are the creation of knowledge taxonomy and the creation and storage of the experience base. This could be achieved by using cloud based database or storage cache.

Workflow management is responsible for most KM activities: knowledge creation, storage, dissemination and reuse/application. While it is keen to go for total agent-based solution, SDeX’s agents will only focus on the critical parts: the experience recommendation, dissemination, reuse and statistical update. This is to ensure that the system is not burdened with ‘too’ many agents which may affect the system performance.

5.2. Workflow Integration

Fig. 3 presents the workflow of the main entity of SDeX: the management of the experience. A product experience is first created by a project owner. It is then opened to the project contributors for the creation of experiences. Creating or editing experiences is a cyclic process happens during the project lifecycle until it is completed. Completed product experiences are submitted for dissemination approval. Project owner verify the product experience and proceed with approval process. If it is approved, the product experience is published and disseminated to the users. If it is not, it is sent back to the contributor for revision or amendment. A process workflow is responsible to handle the approval of the process at the background. Additionally, the workflow will also handle the approval of user role when there are requests for project contributor or subscriber.

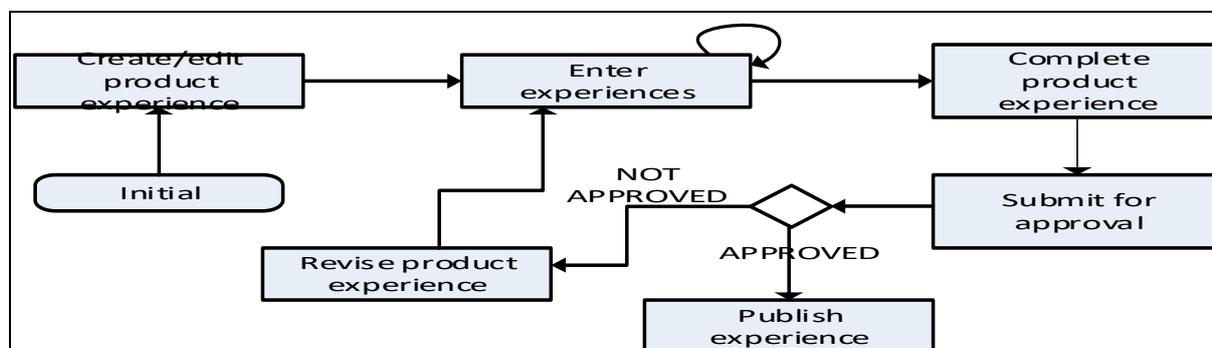


Fig.3. Experience management process workflow

5.3. Multi-Agent System in SDeX

In SDeX, the intelligent business process is handled by workflow and autonomous agents. As shown in Fig. 4, there are three co-operative agents defined: Recommendation Agent, Experience Agent and Statistics Agent.

A Recommendation Agent, detects profiles that are still not subscribed to a particular content that are relevant to his/her interest and/or expertise. The agent further communicates with Experience Agent via content querying protocol to screen all the content from the experience base and compare, if there are still contents that are not subscribed by particular user profiles based on their registered interests and/or expertise. In such a case, the agent alerts the user spontaneously to subscribe to that content. This is to increase user awareness and further maximize the usage of the existing experiences and expose reuse potential.

An Experience Agent's role is to perceive any incoming Published experience. When an experience is ready for publication, by using profile querying protocol, the agent finds the relevant profiles. The matched profiles include project owner and contributor as well as other SDeX users who have similar interests or have subscribed to the project or topic.

A Statistics Agent is responsible to provide information on the current practice graphically to allow organizations to gain insight on the current practices of the community and allow them to react and feedback accordingly. Reports like effort spent in each phase, efficiency on the techniques used, efficiency and productivity of sprints (in case of Scrum projects) will be made available. Additionally, this agent will also receive information on any Approved, Published, or Published_and_Reuse. Statistics update protocol will evaluate the information received from an Experience Agent on the states of the deposited experiences and will update the representation of the experience statistics, e.g. chart or gauge meter on the user interface directly. The goal of the statistical update is to engage the user and to keep them informed and aware of the available experiences in SDeX. Besides increasing the awareness among SDeX users, this will also draw attention from other public users of the organization [29].

The specification and design of MAS can be done using Prometheous methodology [36], while the implementation of software agent can be written in any kind of language. There are a number of agent frameworks that have been introduced by previous researchers. Some of the most interesting are Cougaar, JACK, 3APL and Agent Factory, JADE. A review of these agent implementations is available in [30]. Some points to note from the analysis are Cougaar, JACK and JADE are based on Java language; JACK and Agent Factory are BDI-based (Belief-Desire-Intention), and JADE is an agent platform based on FIPA compliance. FIPA

(Foundation for Intelligent Physical Agents) specifications represent a collection of standards, which are intended to promote the interoperation of heterogeneous agents and the services that they can represent. JADE’s goal is to simplify development while ensuring standard compliance through a comprehensive set of system services and agents [31]. The usage of JADE’s framework for the agent development is still under evaluation.

6. CONCLUSION

The importance of knowledge retention in SD process has led to the development of this experience base factory model. A model has been formulated earlier and eventually the prototype called SDeX have been designed, which is the main focus in this paper. The main functionality of the system is to provide a platform to store all experiences gained during the development of the software in a structured forms, which is based on defined software process models.

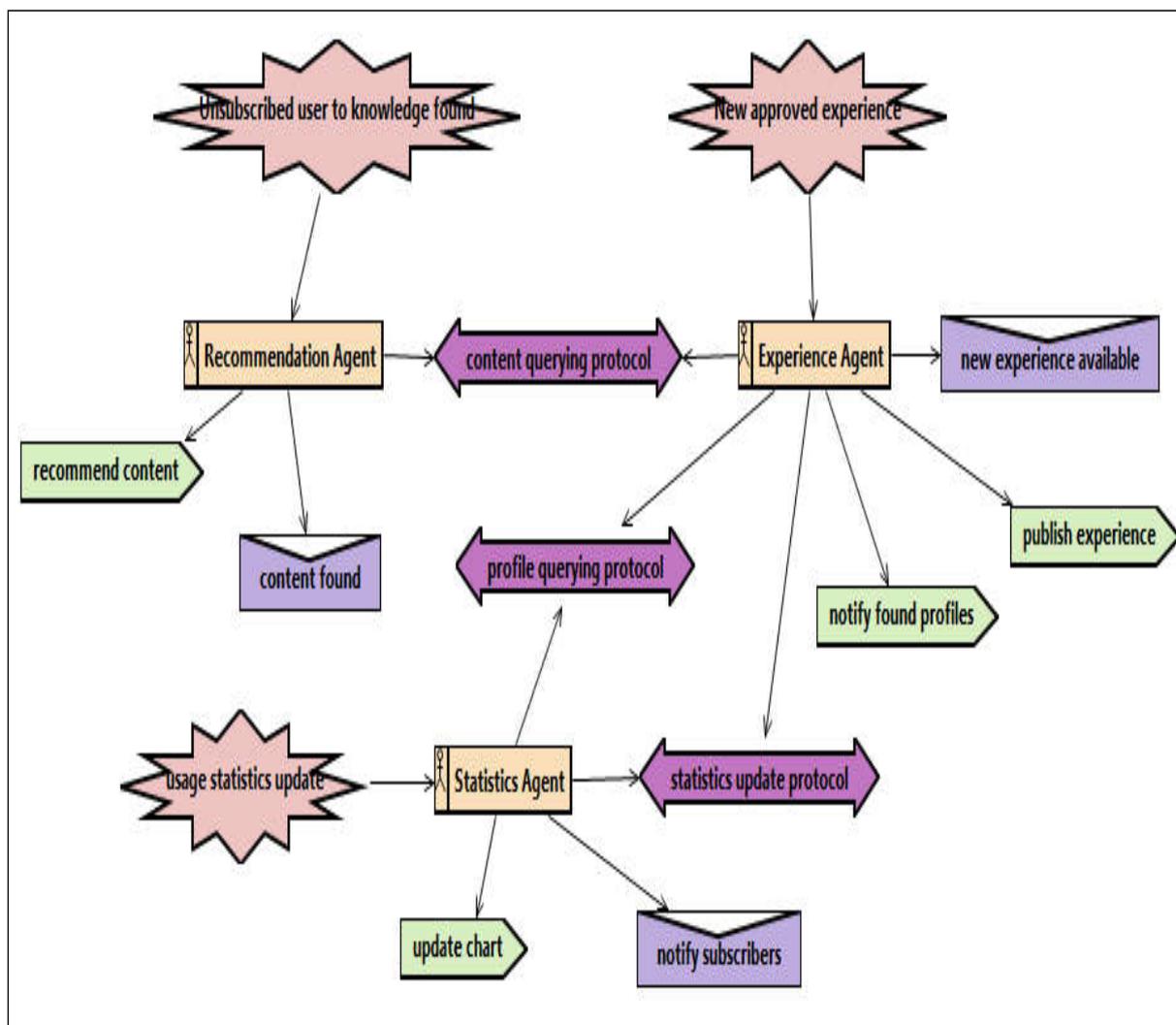


Fig.4. SDeX agent overview diagram

The outcome from the experience factory is reusable product experiences that could be reused in another projects in an attempt to allow quicker and higher quality product development as well as to enhance organizational learning. Additionally, in order to ensure greater usage of the application, automation elements are incorporated into the system with the aid of workflow, multi-agents and cloud computing environment. It is expected that by incorporating adequate business process automation, SDeX effectiveness and efficiency can be achieved without burdening the system performance. The application will also be able to provide some insights on the current practices such as efficiency and productivity that will allow organizations and the community to have continuous feedback loop for organizational improvement. In the future study, the prototype will undergo evaluation phase in which the effectiveness and efficiency of the model will be measured.

7. REFERENCES

- [1] Hanafiah M, Abdullah R, Din J, Murad M A. Towards developing collaborative experience based factory model for software development process in cloud computing environment. *International Review on Computers and Software*, 2015, 10(3):340-350
- [2] Rocha R G, Azevedo R, Meira S. A proposal of an ontology-based system for distributed teams. In *40th IEEE EUROMICRO Conference on Software Engineering and Advanced Applications*, 2014, pp. 398-401
- [3] Salger F, Engels G. Knowledge transfer in global software development: leveraging acceptance test case specifications. In *32nd ACM/IEEE International Conference on Software Engineering*, 2010, pp. 211-214
- [4] Ardimento P, Cimitile M, Visaggio G. Distributed software development with knowledge experience packages. In Y. T. Demey, & H. Panetto (Eds.), *OTM confederated international conferences" on the move to meaningful internet systems*, Berlin: Springer, 2013, pp. 263-273
- [5] Stapel K, Schneider K. Managing knowledge on communication and information flow in global software projects. *Expert Systems*, 2014, 31(3):234-252
- [6] Wende E, Philip T, Schwabe G, King G. KAIWA: Towards a method for knowledge transfer in the transition phase of offshore outsourced projects. In *International Workshop on Global Sourcing of Information Technology and Business Processes*, 2013, pp. 180-191
- [7] Monte-Alto H, Biasão A, Teixeira L, Huzita E. Multi-agent applications in a context-aware global software development environment. In S. Omatu, J. F. D. P. Santana, S. R. González, J. M. Molina, A. M. Bernardos, & J. M. C. Rodríguez (Eds.), *Distributed*

computing and artificial intelligence: 9th international conference. Berlin: Springer-Verlag, 2012, pp. 265-272

[8] Ivarsson M, Gorschek T. Tool support for disseminating and improving development practices. *Software Quality Journal*, 2012, 20(1):173-199

[9] International Society of Automation (ISA). What is automation? North Carolina: ISA, 2017

[10] Lai J, Fan Y. Workflow and knowledge management: Approaching an integration. In Y. Han, S. Tai, & D. Wikarski (Eds.), *Engineering and deployment of cooperative information systems*. Berlin: Springer-Verlag, 2002, pp. 67-71

[11] Caverlee J, Bae J, Wu Q, Liu L, Pu C, Rouse W B. Workflow management for enterprise transformation. *Information Knowledge Systems Management*, 2007, 6(1, 2):61-80.

[12] Jennings N R, Norman T J, Faratin P, O'Brien P, Odgers B. Autonomous agents for business process management. *Applied Artificial Intelligence*, 2000, 14(2):145-189

[13] Wooldridge M, Jennings N R. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 1995, 10(2):115-152

[14] Talib A M, Atan R, Abdullah R, Murad M A. Multi agent system architecture oriented Prometheus methodology design to facilitate security of cloud data storage. *Journal of Software Engineering*, 2011, 5(3):78-90

[15] Toledo C M, Chiotti O, Galli M R. A multi-agent knowledge management system for reactive and proactive knowledge supply. In P. O. de Pablos (Ed.), *Knowledge management and drivers of innovation in services industries*. Pennsylvania: IGI Global, 2012, pp. 203-218

[16] Nor M Z, Abdullah R, Selamat M H, Murad M A. An agent-based Knowledge Management System for collaborative software maintenance environment Design and evaluation. In *IEEE International Conference on Information Retrieval and Knowledge Management*, 2012, pp. 115-120

[17] Mari M, Poggi A, Tomaiuolo M. Agent-based network infrastructure for e-communities. In G. D. Putnik (Ed.), *Encyclopedia of networked and virtual organizations*. Pennsylvania: IGI Global, 2008, pp. 10-18

[18] Talib A M, Atan R, Abdullah R, Murad M A. Towards new data access control technique based on multi agent system architecture for cloud computing. *Communications in Computer and Information Science*, 2011, 189(Part 2):268-279

[19] Odell J. Agents: A necessary ingredient in today's highly collaborative world. *Business Technology Trends and Impacts, Council Opinion*, 2007, 8(1):

-
- [20] Moldt D, Quenum J, Reese C, Wagner T. Improving a workflow management system with an agent flavour. In International Workshop on Petri Nets and Software Engineering, 2010, pp. 55-70
- [21] Ma Z. Managing big data in cloud computing environments. Pennsylvania: IGI Global, 2016
- [22] Badger L, Grance T, Patt-Corner R, Voas J. Draft cloud computing synopsis and recommendations-Recommendations of the National Institute of Standards and Technology. NIST Special Publication 800-146, Maryland: National Institute of Standards and Technology, 2011
- [23] Langmann R, Meyer L. Automation services from the cloud. In 11th IEEE International Conference on Remote Engineering and Virtual Instrumentation, 2014, pp. 256-261
- [24] Ramakrishnan M. Oracle process cloud service: Rapid business process automation in the cloud. California: Oracle, 2015
- [25] Mudialba PJ. The impact of cloud technology on the automation of businesses. In IEEE International Conference on Platform Technology and Service, 2016, pp. 1-4
- [26] Hanafiah M, Abdullah R, Murad M A, Din J, Nor M Z. Experience based factory model for software development process: Item construct validation on questionnaire design. Journal of Theoretical and Applied Information Technology, 2017, 95(1):177-195
- [27] Google Cloud Platform. AngularJS + cloud endpoints-A recipe for building modern web applications. 2016, <https://cloud.google.com/solutions/angularjs-cloud-endpoints-recipe-for-building-modern-web-applications>
- [28] TechTarget. OAuth. 2012, <http://searchmicroservices.techtarget.com/definition/OAuth>
- [29] Kamble D R. Architectural review on multi agent knowledge management. International Journal of Scientific and Technology Research, 2013, 2(6):105-114
- [30] Odell J, Nodine M. The foundation for intelligent physical agents. 2006, <https://www.fipa.org/docs/input/f-in-00085/f-in-00085.pdf>
- [31] Bellifemine F, Poggi A, Rimassa G. JADE-A FIPA-compliant agent framework. In Practical Applications of Agents and Multi-Agent Systems, 1999, pp. 97-108
- [32] Hall G. Pearson's correlation coefficient. Other Words, 2015, 1(9):1-4
- [33] Talib A M, Atan R, Abdullah R, Murad M A. Security framework of cloud data storage based on Multi Agent system architecture-A pilot study. In IEEE International Conference on Information Retrieval and Knowledge Management, 2012, pp. 54-59

-
- [34] Jabar S, Khalefa M A, Abdullah M S, Abdullah R H. Meta-analysis of ontology software development process. *International Review on Computers and Software*, 2014, 9(1):29-37
- [35] Zhao Y, Dong J, Peng T. Ontology classification for semantic-web-based software engineering. *IEEE Transactions on Services Computing*, 2009, 2(4):303-317
- [36] Padgham L, Winikoff M. Prometheus: A methodology for developing intelligent agents. In *ACM 1st International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1*, 2002, pp. 37-38.

How to cite this article:

Hanafiah M, Abdullah R, Din J, Murad MAA. Architectural design of experience based factory model for software development process in cloud computing: integration with workflow and multi-agent system. *J. Fundam. Appl. Sci.*, 2017, 9(4S), 210-225.