International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

Vol. 34 No. 2 September 2022, pp. 150-158

A Framework For Loosely Coupled Components To Automate Municipal Services (FLCCAMS)

Mohammed M I Awad

Faculty of Information Technology
University of Palestine
Gaza, Palestine
dr.awad.1981@gmail.com

***Abstract*— Unifying and computerizing the procedures of municipal operations and maintenance is of great importance to municipalities as well as to ministries of local governments. On top of that, enhancing municipal e-services in terms of efficiency, quality, reusability, distribution and interoperability of these services; is a goal for most of municipalities. Municipalities need to unify and computerize the procedures of maintenance and operations for their assets. This will eliminate the difficulties to prioritize the maintenance tasks; to let every task takes its turn fairly. In addition, this will speed up performing such maintenance and operation tasks, and will decrease the bugs appear after the maintenance is done. Therefore, the objective of this research work is to define, design and evaluate a conceptual framework for loosely coupled components to automate municipal services as well as unifying and computerizing maintenance and operations in local governments. The scope of this research covers roads and public buildings as the most important parts of municipal assets.**

***Keywords*— Municipal services; Reusability; Distribution; Interoperability; loose coupling**

I. Introduction

The use of distributed systems for ministries of local government as well as for municipal applications is highly recommended due to the following motivations [1-4]:

A. Access to geographically remote data and resources

In many scenarios, the data cannot be replicated at all sites participating in the distributed execution because it may be too large or too sensitive to be replicated. For example, data within a ministry of local government that is common for all municipalities nationwide; is both too large and too sensitive to be replicated at every municipality. It is therefore stored at a central server that can be queried by branch offices. In addition to that, limited resources in mobile devices as well as in the wireless technology with which these devices communicate have increased the need for distributed protocols and middleware.

B. Enhanced reliability

The distribution concept has increased the system reliability because of the possibility of replicating resources and executions, and since geographically distributed resources are not likely to crash at the same time under normal circumstances. Reliability is dependent on several aspects such as:

- Availability: The resource should be accessible at all times.
- Integrity: The value/state of the resource should be correct, in the face of concurrent access from multiple processors, as per the semantics expected by the application.

- Fault-tolerance: The ability to recover from system failures, where such failures may be defined to occur in one of the many failure models.

C. *Increased performance/cost ratio*

By resource sharing and accessing geographically remote data and resources, the performance/cost ratio is increased. Although higher throughput has not necessarily been the main objective behind using a distributed system, nevertheless, any task can be partitioned across the various computers in the distributed system. Such a configuration provides a better performance/cost ratio than using special parallel machines.

D. *Scalability*

As the processors are usually connected by a wide-area network, adding more processors does not pose a direct bottleneck for the communication network.

E. *Modularity and incremental expandability*

Heterogeneous processors may be easily added into the system without affecting the performance, as long as those processors are running the same middleware algorithms. Similarly, existing processors may be easily replaced by other processors.

II. Gap and Problems

Municipalities have many functions such as Land development, water and sewer, roads, cemeteries, buildings, waste management and fire-fighting. The failure of these services could put the urban life into problems, which is the responsibility of the local government or municipality of a certain city in a certain country. In addition, it is the responsibility of the municipalities to build, maintain and repair roads, buildings, squares and parks [5-7].

Many municipalities worldwide still use manual and traditional methods in serving the citizens, and in prioritizing the tasks need to be done. Most of these municipalities recognize the important effects got by enhancing their IT infrastructure and automating these services improve their ability to deliver services and manage their operations [8-10]. Some municipalities already adopt new technologies, while others have been slow to start automating their operations. Roads and buildings are two of the major fields of the municipal work. These two fields need maintenance services frequently. In addition, they need to be operated. Therefore, there is a need for a research work that analyzes the maintenance services for the roads and buildings at the municipalities. Some municipalities have already used computer applications for this purpose through different applications such as Municipal Management Software and others. These software applications are sometimes powerful, easy to use and modular in design [11-13]. On the other hand, these applications are general and do not touch the details needed to handle maintenance and operation activities for roads and buildings, and more important, most of those applications do not adopt distribution and interoperability concepts [14-16].

III. The Framework For Loosely Coupled Components To Automate Municipal Services

To overcome the common problems that municipalities face during their maintenance and operations processes in the roads and buildings field; this paper defines a framework that unifies and automats maintenance operations in local governments and/or municipalities. FLCCAMS framework unifies and automates these processes of operation and maintenance through workflow processes. These processes are distributed, loosely-coupled, interoperable and scalable to cover the municipalities' other assets in the future. In addition, the model supports the availability of a database for the municipality fixed assets which is the base of municipalities' operations. Following such solution, a municipality should be able to control the efficiency and performance of each fixed asset and follow the entire lifecycle from acquisition to disposal.

Expenses are major issues in this framework. FLCCAMS allows the identification of any expenses into two types (revenue expenses and capital expenses) and reflects the capital expenses to the original value of asset, age of asset and rate of depreciation. Furthermore, FLCCAMS includes the necessary awareness of having Application Program Interfaces (APIs) for the purpose of integration with other legacy systems implemented at the municipalities.

For automating the unified framework, "fixed assets registration" is a base process in the conceptual model; it provides a database for the municipality fixed assets with all asset attributes.

The framework allocates the additional information required for integration procedures with other legacy systems at a certain municipality. For instance, if this legacy system is a financial system, it should identify the asset belong to which budget, and belong to which cost center. In addition, it identifies the revenue expenses and capital expenses and reflects the capital expenses to the original value of asset, age of asset and the depreciation rate.

A certain asset could be divided to one or more segments. Each segment contains elements. Figure 1 depicts the

process of adding a new asset and segment with their elements. In addition, figure 2 depicts the process of maintenance plan preparation.

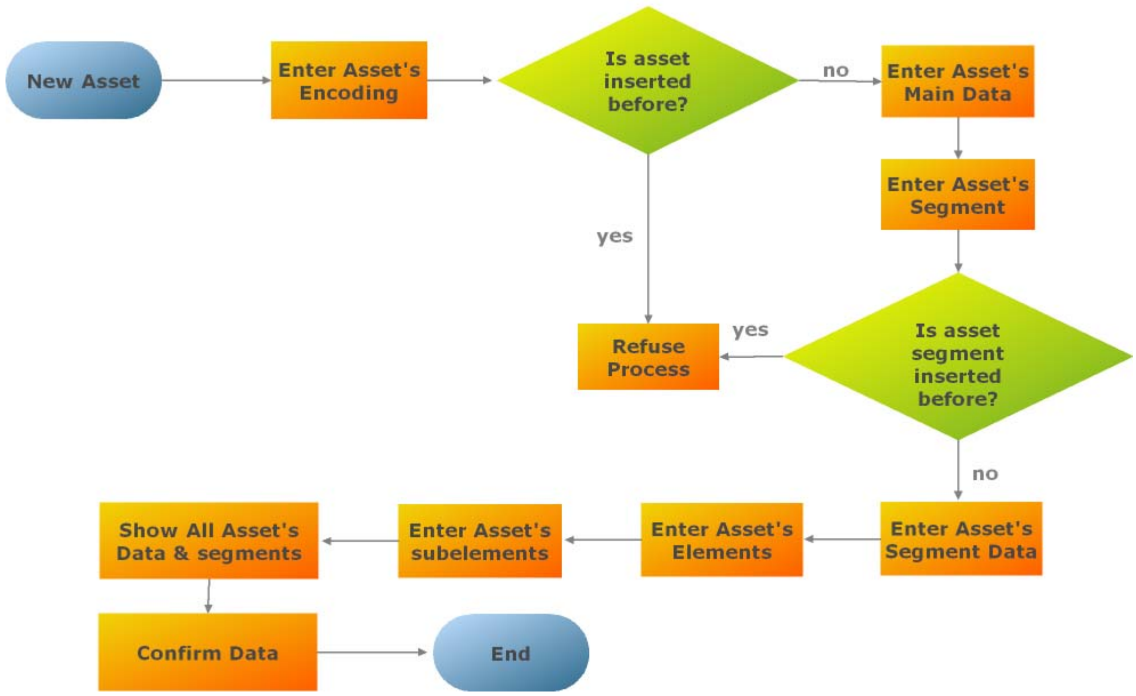

Fig. 1. The process of adding a new asset and segment with their elements

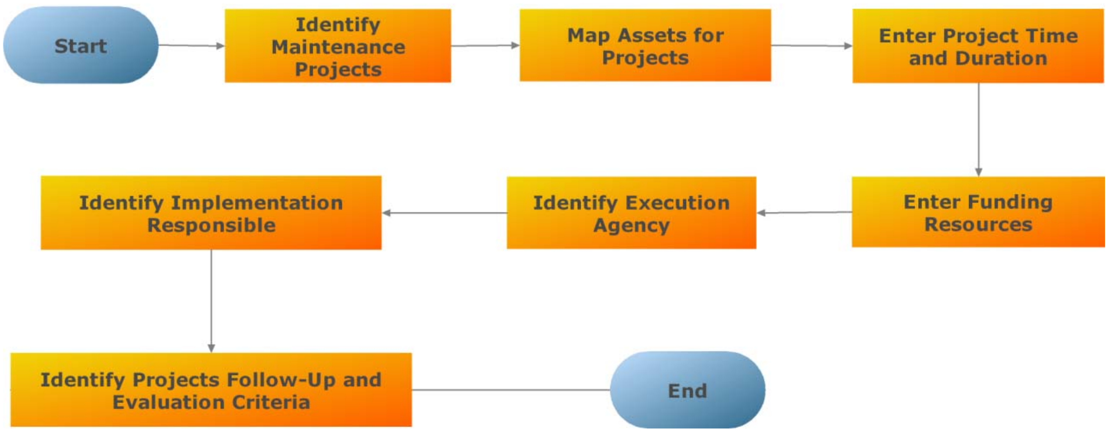

Fig. 2. The process of maintenance plan preparation

The maintenance procedures are implemented and executed according to a planned schedule. A work orders dataset will be generated for each maintenance execution in order to allow the responsible person or department to follow-up the execution according to the approved plan. Figure 3 depicts the process of maintenance plan execution.

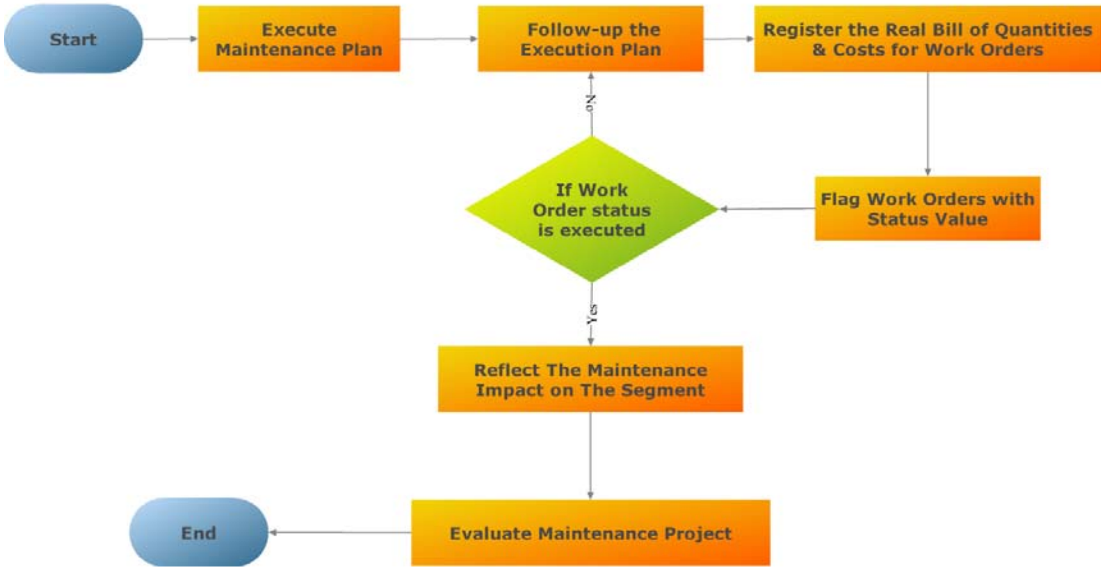

Fig. 3. The process of maintenance plan execution.

As explored in figures 1, 2 and 3, the FLCCAMS provides unified processes for municipalities which aim to enhance and automate the maintenance processes of roads and buildings within a systematic workflow process. These processes are scalable to cover the municipal other assets in the future. Using FLCCAMS, a certain municipality can control the efficiency of each fixed asset and follow the entire lifecycle from acquisition to disposal.

In addition to the theoretical model, this paper suggests a technical architecture that could be the base for developing a system based on this theoretical model. This architecture defines a structured solution and required processes taking in consideration technical and operational needs to fulfill business and functional needs.

As a popular reliable distribution and interoperability architecture, several benefits can be obtained when SOA is used in designing the municipal components within a distributed interoperable framework. These benefits include: distribution, interoperability, reusability, portability, and compatibility with legacy systems [7, 17].

Since one of the major roles of SOA is to enhance the features of computerized frameworks and models, and this research's framework is a computerized framework, therefore, SOA can play a central role in enhancing the framework features regarding component distribution and interoperability, section III explores how SOA is adopted to achieve distribution and interoperability for FLCCAMS components.

IV. The Framework Specifications

Since For figures 1, 2 and 3, every figure represents a separate component of the FLCCAMS framework. These 3 figures (FLCCAMS components) should be distributed and interoperable. SOA (Service Oriented Architecture) is adopted to structure the FLCCAMS framework. As a result of that, FLCCAMS has been empowered by several additional features in addition to distribution and interoperability [7, 17, 18]. These features are:

A. Portability and Interoperability

The portability feature, which SOA adds to its services and which is a direct result to XML adoption for data exchange between the web services leads to that any of the framework components can be used as a portlet in a web portal. These can also be remotely managed through the web portal or even a normal web application to centralize the management of all the components even if they are available in different locations.

B. Compatibility

As a result of applying SOA, any component of the framework can be compatible with any platform or development environment. For example, the “maintenance project identification” component as well as any other component can be compatible with .NET or JavaEE technologies without any change in the component coding. This is a direct result to the compatibility feature which SOA framework offers.

C. Platform-neutrality

XML-based message information flow enhances the capability to achieve platform neutrality. These XML messages are based on XML standards, which eliminate the need to set up other messaging standards that can differ across platforms between the framework components. For example, if a “municipality building operation” component is encapsulated in a Java service inside a J2EE framework, and another component is encapsulated in a .NET service inside a .NET framework, the two services can exchange messages while each is available in a different framework.

D. Extensibility and Reusability

One of the most important goals of adapting SOA is the component extensibility as well as reusability. Traditional municipal software components are tightly coupled, which have leaded to difficulties to extend and reuse the software components in heterogeneous environments.

As described above, it is clear that SOA is adopted for the abstract structure of the main components of the FLCCAMS framework without touching the entire details of each component.

The entire details and processes of the 3 components are explored in figures 1, 2 and 3. For those very specific details and processes of each component, the recommended design pattern to be used for the development environment is the MVC (Model-View-Controller) design pattern. This could be achieved through many different frameworks. For example, one of these is the CodeIgniter framework that is based on the MVC design and development pattern. MVC is a software approach that separates application logic from presentation. It permits the web pages to contain minimal scripting since the presentation is separate from the PHP scripting (in case of using PHP). The MVC pattern that is recommended to be applied for implementing the details and processes of FLCCAMS components; is shown in figure 4.

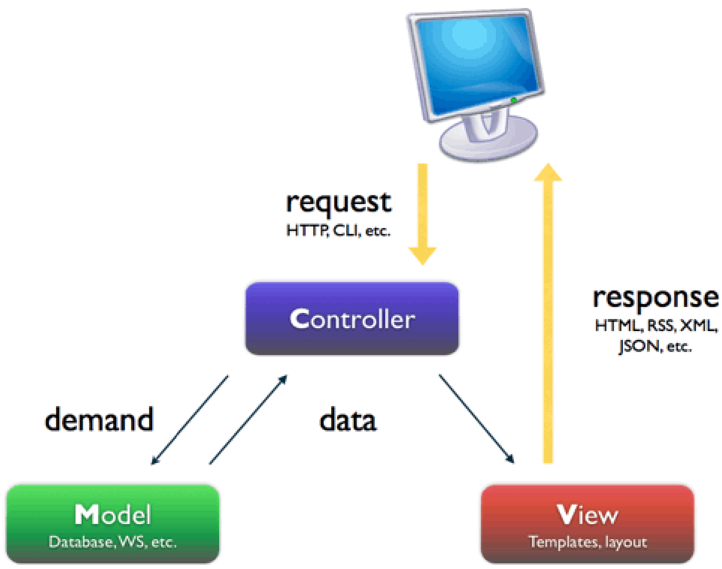

Fig. 4. MVC Design Pattern for implementing FLCCAMS

As shown in figure 4, MVC divides the coding into three main parts, these are: Model, View and Controller.

The *Model* represents FLCCAMS components' data structures. Typically, each component classes will contain functions that help to retrieve, insert, and update information in the DBMS.

The *View* is the information that is being presented to the municipalities' end users. A View will normally be a web page, in case of using CodeIgniter as an MVC example; a view can also be a page fragment like a header or footer. It can also be an RSS page, or any other type of "page".

The Controller serves as an intermediary between the Model, the View, and any other resources needed to process the HTTP request and generate a web page.

More flexibility is given to the FLCCAMS by MVC design pattern implementations such as CodeIgniter. MVC has a fairly loose approach when Models are not required. If future developers need to expand the implementation of FLCCAMS framework, and they don't need the added separation between Models and Controllers, they can ignore them and build the new parts minimally using Controllers and Views. MVC also enables the developers to incorporate their own existing scripts, or even develop core libraries for the system, enabling them to work in a way that makes the most sense to them.

In addition to this description, figure 5 illustrates how data flows throughout the framework (taking PHP as an example).

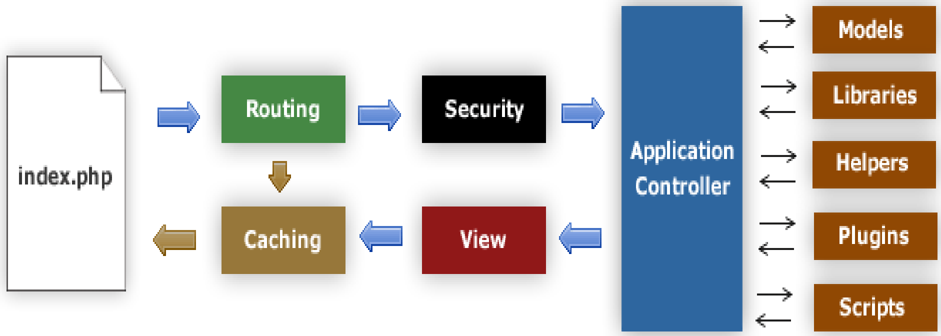

Fig. 5. Data flows throughout the framework

As shown in figure 5:

- The index.php serves as the front controller, initializing the base resources needed to run the implementation of the framework.
- The Router examines the HTTP request to determine what should be done with it.
- If a cache file exists, it is sent directly to the browser, bypassing the normal system execution.
- Before the application controller is loaded, the HTTP request and any user submitted data is filtered for security.
- The Controller loads the model, core libraries, helpers, and any other resources needed to process the specific request.
- The finalized View is rendered then sent to the web browser to be seen. If caching is enabled, the view is cached first so that on subsequent requests it can be served.

V. Recommendations For Realizing The Framework

This section summarizes the recommendations for developing municipal systems as a realization and implementation to the FLCCAMS framework of this research. These recommendations are:

- A distributed project is recommended to be created separately for each component of the system. The main three components are explored in figures 1, 2 and 3. Multi-tier architecture is recommended to be adopted as a base for these projects. The distribution concept could be based on any distributed technology such as the EJB (Enterprise Java Beans) technology as an example, which was built on CORBA and RMI frameworks. GlassFish as an example to application servers could be selected to host the projects and the services of the distributed modules, or any application server that suits the distributed and programming technologies used.

- After creating a distributed project for each component, a web service is recommended to be created inside each project to wrap the business logic of the system task.
- SOA is recommended to be adopted for the general abstract components of the system as described in section III.
- As an implementation to the orchestration point that is a mandatory specification in SOA, the BPEL (Business Process Execution Language) is an example to implement the orchestration point of the system. The BPEL has rapidly been emerged as a standard for combining a set of services into a number of discrete and long running enterprise processes. Most organizations who adopt SOA are either using BPEL or planning to use it over their other middleware framework.
- MVC design pattern is recommended to be adopted for the details and processes for each component of the system as described in section III.

VI. Evaluation

The evaluation was done to measure experts' satisfaction of the FLCCAMS framework to substantiate that the features and specifications of the framework can solve real problems at the municipalities. Eight (8) experts were chosen to answer the questionnaire prepared for this evaluation process. Since the evaluation is used to measure much specified dimensions of the FLCCAMS framework, the evaluators were selected from the municipal and IT fields. Figure 6 shows the mean of the level of satisfaction of the experts. The evaluation process included two dimensions; the first is that the features and specifications of the framework can solve real problems at the municipalities, while the other is that the framework can easily be realized to have a system that automates the maintenance processes at the municipalities. Overall 89% of the experts are satisfied with it. They rated 24% as "strongly agree" and 65% as "agree", while overall 6% of the experts have "neutral" opinions. Nevertheless, 5% of the experts are unsatisfied with the FLCCAMS framework.

Scale	Total No. of Experts' hits	Mean of Experts	Acceptance Rate
1:strongly disagree	0	0	0.00%
2:disagree	6	0.083	1.67%
3:neutral	29	1.375	27.50%
4:agree	78	3.375	67.50%
5:strongly agree	7	0.167	3.30%

Fig. 6. Responses details of the experts

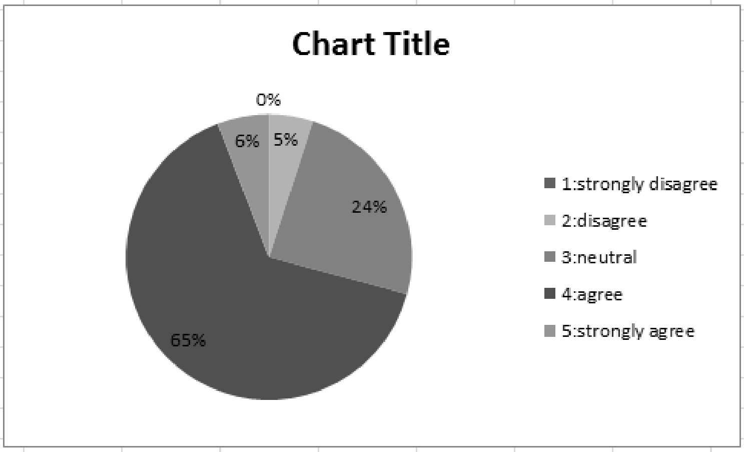

Fig. 7. The mean of the level of satisfaction of the experts

VII. Conclusions

This research work is conducted to overcome the problems resulted from lack of distribution and interoperability of the available municipal systems; in addition to resolving prioritizing problems and handling maintenance issues at local governments or municipalities. The research has come out with a conceptual distributed interoperable framework for unifying and computerizing maintenance operations in local governments. In addition, this framework (FLCCAMS) has recommended a technical architecture to be followed within the system development based on the FLCCAMS framework. The feedback of experts on the model has shown an accepted satisfactory level. Therefore, this paper presents a framework to be followed for unifying and computerizing maintenance operations in local governments.

References

[1] B. Burns, *Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services.* " O'Reilly Media, Inc.", 2018.

[2] S. Gholamian and P. A. Ward, "What Distributed Systems Say: A Study of Seven Spark Application Logs," in *2021 40th International Symposium on Reliable Distributed Systems (SRDS)*, 2021, pp. 222-232: IEEE.

[3] Z. S. Ageed, R. K. Ibrahim, and M. Sadeeq, "Unified ontology implementation of cloud computing for distributed systems," *Current Journal of Applied Science and Technology,* vol. 39, pp. 82-97, 2020.

[4] G. Audrito, R. Casadei, F. Damiani, G. Salvaneschi, and M. Viroli, "Functional programming for distributed systems with XC," in *36th European Conference on Object-Oriented Programming (ECOOP 2022)*, 2022: Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[5] C. A. Bana and R. C. Oliveira, "Assigning priorities for maintenance, repair and refurbishment in managing a municipal housing stock," *European Journal of Operational Research,* vol. 138, no. 2, pp. 380-391, 2002.

[6] J. Mongkolnchaiarunya, "Promoting a community-based solid-waste management initiative in local government: Yala municipality, Thailand," *Habitat International,* vol. 29, no. 1, pp. 27-40, 3// 2005.

[7] Y. Zhou, Y. Ye, and W. Tang, "Construction and application on administrative service management system of Shanghai Municipal Agriculture and Rural Affairs Commission based on SOA," *Acta Agriculturae Shanghai,* vol. 35, no. 6, pp. 141-147, 2019.

[8] Å. Johnsen, P. Meklin, L. Oulasvirta, and J. Vakkuri, "Performance auditing in local government: an exploratory study of perceived efficiency of municipal value for money auditing in Finland and Norway," *European Accounting Review,* vol. 10, no. 3, pp. 583-599, 2001/09/01 2001.

[9] J. K. Scott, "“E” the People: Do U.S. Municipal Government Web Sites Support Public Involvement?," *Public Administration Review,* vol. 66, no. 3, pp. 341-353, 2006.

[10] F. M. d. Silva, A. C. Menks, A. L. C. Prudente, J. C. L. Costa, A. E. M. Travassos, and U. Galatti, "Squamate Reptiles from municipality of Barcarena and surroundings, state of Pará, north of Brazil," *2016,* vol. 7, no. 3, p. 7, 2016-12-13 2016.

[11] S. Dury, J. Willems, N. De Witte, L. De Donder, T. Buffel, and D. Verté, "Municipality and Neighborhood Influences on Volunteering in Later Life," *Journal of Applied Gerontology,* vol. 35, no. 6, pp. 601-626, 2016/06/01 2014.

[12] T. H. Poister and G. Streib, "Performance Measurement in Municipal Government: Assessing the State of the Practice," *Public Administration Review,* vol. 59, no. 4, pp. 325-335, 1999.

[13] S. Selvam, F. A. Dar, N. S. Magesh, C. Singaraja, S. Venkatramanan, and S. Y. Chung, "Application of remote sensing and GIS for delineating groundwater recharge potential zones of Kovilpatti Municipality, Tamil Nadu using IF technique," *Earth Science Informatics,* journal article vol. 9, no. 2, pp. 137-150, 2016.

[14] C. A. Bana e Costa and R. C. Oliveira, "Assigning priorities for maintenance, repair and refurbishment in managing a municipal housing stock," *European Journal of Operational Research,* vol. 138, no. 2, pp. 380-391, 4/16/ 2002.

[15] B. Dollery and A. Johnson, "Enhancing Efficiency in Australian Local Government: An Evaluation of Alternative Models of Municipal Governance," *Urban Policy and Research,* vol. 23, no. 1, pp. 73-85, 2005/03/01 2005.

[16] R. Saberifar, "Determination and Identify Factors Influencing in Designing an Smart Organization Model for Urban Management (Case study: Municipality of Mashhad)," *Geographical Urban Planning Research (GUPR),* vol. 8, no. 2, pp. 445-467, 2020.

[17] N. Niknejad, W. Ismail, I. Ghani, B. Nazari, and M. Bahari, "Understanding Service-Oriented Architecture (SOA): A systematic literature review and directions for further investigation," *Information Systems,* vol. 91, p. 101491, 2020.

[18] M. Fischer, F. Imgrund, C. Janiesch, and A. Winkelmann, "Directions for future research on the integration of SOA, BPM, and BRM," *Business Process Management Journal,* 2019.