Developing Collaborative Applications with Actors

Saraí Gallardo-Vera, Eric Nava-Lara.

***Abstract*—We present an actor-role based framework that allows risk assessment process optimization for a safe workplace in a power plant. This framework was designed using a service-oriented approach and was implemented as a software collaboration tool using different web application technologies.**

***Index Terms*—Actor, software framework, evaluation list, maximum probable loss.**

I. Introduction

THE safety of power plants needs a comprehensive system to send and display activities and risks exposed to personnel, as well as goods and production processes in order to determine the maximum probable loss of energy in the buildings. In many cases, these risk assessments are made on paper by the staff, sending this way the resulting message of the assessment list. Moving from a paper based to an automatized process an appropriate component and service design is required, as well as agile programming models.

Message-passing is the most attractive solution because it is a concurrent model not based on data sharing so its techniques can be used in distributed computation too. One of the well-known theoretical and practical models of message-passing is the *actor model* [1]. Using this approach, programs become collections of independent active objects (actors) that exchange messages and have no mutable shared state. Actors can help developers avoid issues such as deadlock, live-lock and starvation, common problems of shared memory based approaches. There are a multitude of actor-oriented libraries and languages, each of them implementing some variants of actor semantics. However, such libraries and languages use either thread-based programming, which makes the program development easy, or event-based programming, far more practical to develop large and efficient concurrent systems, but also more difficult to use [2].

Manuscript received July 10, 2015; revised July 20, 2015.

This work was supported by the System of Level Risk Evaluation (SENS), used in power plants of the Federal Electricity Commission (CFE), México.

Saraí Gallardo-Vera is a researcher of the Electric Research Institute (IIE). Cuernavaca, Morelos, México and has vast experience with management processes and corporate portals (e-mail: sgallardo@iie.org.mx).

Eric Nava-Lara is a researcher and project management of the Electric Research Institute (IIE). Cuernavaca, Morelos, México and develops risk management and compliance systems (e-mail: ernava@iie.org.mx).

This paper presents an actor-role framework with risk level evaluation, called LeSRi (Level of Security of Risk), which has the suitable features for both, calculating maximum probable loss of energy and risk levels. Next section introduces related work. Section 3 describes the software framework. Section 4 details the features that make LeSRi suitable for using actor-role systems in different types of services and present the experimentation results. Finally, section 5 concludes this paper by discussing its main features and directions of future work.

II. Related Work

Several services and actor-role oriented components have been proposed over the last few decades, a large part of them use a software framework design. In the web service composition area different authors have presented a number of service composition frameworks/methods. The rest of the section presents some of the most interesting works.

A. UML for service

Presented by Christophe Dumez et al [3]. They defined the static aspects of the composition i.e. interface of the Service Composition by UML-Class diagram (WSDL interface and data types involved), and used the UML-activity diagram to model the dynamic aspects (The composition scenario itself, i.e. the interaction between the existing services).

B. Skogan

They propose a service composition using the UML activity diagram as a method, a UML profile and transformation rules that can be used to produce UML models of web service compositions that are executable [4]. They have provided a way to model the coordination and the sequencing of the interactions between web services. However, in this approach, methods, input/output and data transformation are modeled as notes (i.e. comments) on the side of the workflow, which can get quite confusing when the composition flow gets complex.

C. Service Oriented Architecture

SOA paradigm makes the application development easy by coupling services over intranet and via the Internet [5]. SOA paradigm has changed the Internet from being a repository of data to a repository of services [6]. SOA is an architectural style in which software applications are comprised of loosely coupled reusable services by integrating these services through their standard interface. Services are independent of language, platform and location and may be locally developed or requested from the provider. A business process can be realized as a runtime

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

orchestration of set of services.

A limitation of all of the aforementioned frameworks is that they do not treat actor-role as a separate service. During the development of this work, actor-role applications are defined along service composition modeling, performed using the diagram at two different stages; however we do not differentiate them as system-level and service-level.

III. Software Framework

LeSRi (Level of Security of Risk), is an actor-role based software framework that has the goal of simplifying the development of complex systems and guaranteeing the efficient execution of applications.

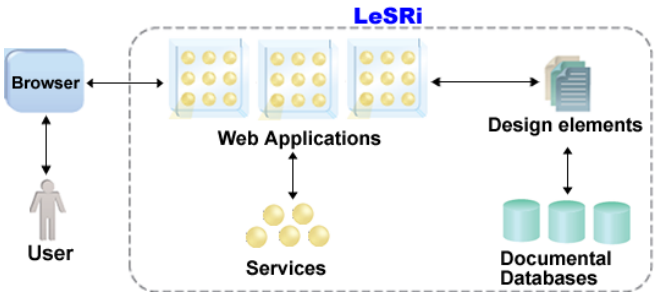

Fig.1. Application architecture.

LeSRi is implemented using the LotusScript language and takes advantage of preexistent jQuery, JSON and Ajax software libraries and solutions for supporting hierarchy and distribution. LeSRi has an architecture composed of an application and a runtime layer. The application layer provides the software components that an application developer needs to extend or directly use for implementing the specific actors of an application. The runtime layer provides the software components that implement the LeSRi middleware infrastructures to support access of standalone and distributed applications.

A. Model View

In LeSRi, an application is based on a set of services that perform tasks concurrently. A service is a collection of forms and views, which interacts with other design elements by exchanging document flow. Moreover, it can create new documents, update its local state, change its elements and send itself [7].

Communication between services and actors is buffered: incoming documents are stored in a document database until the actor is ready to process them. Each actor has a system-wide unique identifier called an actor-role that allows it to be referenced in the application on a transparent way. An actor can create documents only for the applications it has access to, that is, the actors it created and for which received the addresses from other actors. After the creation of documents, an actor can change its state until it terminates itself. Each application has the main duty of creating a set of specific documents through a collection of design elements. Therefore, if a request for a document arrives, the mailbox-actor maintains it until a next behavior is able to process it.

An actor can set a list of facility type and process type concepts and then execute required actions if the template is available. However, if it has no explicit actions to evaluate the risk level list (maximum probable loss and substandard conditions), then executes management and evaluation actions.

Depending on the complexity of the evaluation list and the availability of computing and communication resources, one or more actor-roles can manage the evaluation services. An evaluation service is a space that acts as a container for a set of relevant norm concepts and a necessary list for its execution. In particular, an evaluation list takes advantage of two special services: the power plants safety level and the governing process report. The scheduler manages the concurrent execution of the process evaluation. The service provider enables the customization of a result to perform a new evaluation list. (e.g., to copy an existent evaluation list for the energy process). Fig. 1 shows a graphical representation of the architecture of a LeSRi distributed application.

B. Implementation View

The function of a facility in the power generation market is to enhance understanding of potential property risk exposures and to provide an independent review of the loss-control practices and procedures of the organization [8]. Thus, improving risk management can enhance an insurer's competitiveness and profitability.

In LeSRi, an application is based on a set of services that perform tasks concurrently. A service is a collection of forms and designed views, which interacts with other design elements by exchanging document flow. Moreover, it can create new documents, update its local state, change its elements and send itself by running lotus script stored procedures.

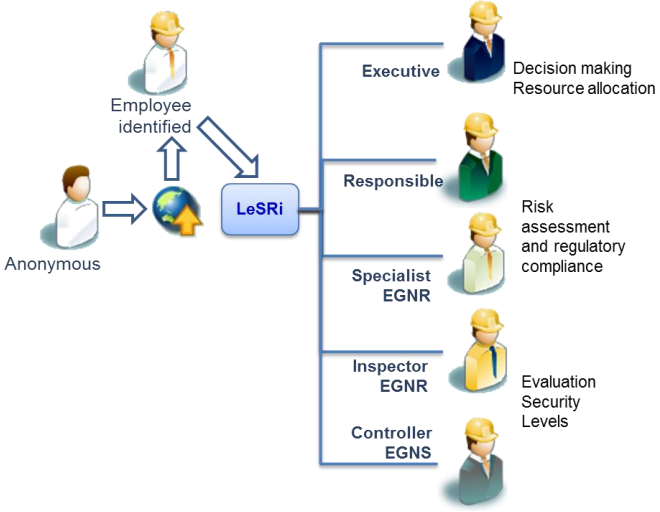

Fig. 2. Actor-role application architecture.

LeSRi provides different actor services and according to designated roles, the use of one or another service influences the attributes of the execution of an application. In particular, actor implementations can be divided in two phases that allows an actor to have its own evaluation list

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

(from here on named level of risk) or to share a single service with other actors of the actor´s space (from here on named evaluation list). Furthermore, the service-specialist of an actor takes advantages of two other main services: collaboration and execution manager. Fig. 2 shows a graphical representation of the architecture of an actor with different roles.

A service supports the distribution of applications to the actor it represents. Therefore, an actor needs to have the specialist-role to create a concepts list. In particular, an actor has the specialist-role if he is assigned an energy facility of another process template, this way either one can create an evaluation list (in fact, the creation method returns the collection of documents of the new evaluation) or it receives an evaluation list from another specialist during execution (each concept of evaluation contains the value of the review) or whose content its result is closed.

A service has an attribute called an organizational-actor, that allows to distinguish itself (and the actor it represents) from the references of other actors of the application that is processing. To guarantee and simplify the implementation, a space acts as a container for the evaluation list running in the same server. In particular, the evaluation list identifier is different for all attributes of spaces of the same specialist-role evaluations.

A mailer service provides a mailbox for the collaboration messages sent to its specialist-role and controller-role until it finishes the evaluation list and delivers messages to other application actors (e.g., the administrator-role). As introduced above, a collaborative service can process a set or specific messages, leaving in the mailbox the messages that it is not able to process. Such messages remain in the mailbox until a new evaluation event is able to process them and if there is no such collaboration they remain in the queue during the actor´s life. A mailbox does not have an explicit limit on the number of messages that can maintain. However, it is clear that the (permanent) deposit of large numbers of messages in the actor´s mailboxes may reduce applications performance and cause in some circumstances their failure.

The actor-role specialist associates a concept with the list processing evaluation task. In LeSRi, a behavior can perform three types of tasks: its qualification, the maximum probable loss and the facilities report. In particular, a behavior does not directly process evaluations, but it delegates the task to some design elements that have the goal of processing the evaluations that match a specific (and irreplaceable) concepts pattern of an applied norm.

Often, an actor´s evaluation list needs to share some information (e.g., a concept of the previous list), this is possible with a template list by process, the type of information that an actor´s evaluation list needs to share depends on the facility type they must perform in an application. Therefore, the state of an evaluation list must be specialized for the task it will perform.

A list of concepts is a collection of documents that contains a set of fields maintaining the value and the substandard conditions by facilities, each list of concepts is different from any other. In fact, the evaluation list of the same specialist-role has different concepts and substandard conditions of different processes with valid results.

A concepts template is a specific list that can apply a combination of qualitative and quantitative values with constraint elements on the result of all the fields of a risk assessment. LeSRi provides a set of constrained predefined templates, but new ones can be easily added. In particular, one constraint allows a pattern assignment to a value of a specific field. Therefore, the addition of field templates will allow the definition of appropriate filters on the values of all the facilities assessment and in particular on the maximum probable loss.

A specialist-role has the duty of supporting the evaluation of its facilities list and enhancing them with new templates types. To do that, an actor takes advantage of some main runtime components (i.e., validations, hierarchy and access control) and of the two special actors: the controller-role and the verifier-service.

The facility has the service of creating reports of the evaluation list details. In particular, it also creates their initial evaluation report.

Finally, the evaluation space provides a runtime component, called configurator-role, which simplifies the configuration of a service by allowing document database access or a procedural method in JAVA (i.e., the PDF writing of executive report files or evaluation reports provided by the administrator-role).

IV. CONFIGURATION GUIDES

The quality of a LeSRi application mainly depends on the execution of the evaluation list and the concepts template. Another important factor that influences its execution is the implementation of the new concepts and the risk factor. However, a combination of such implementation, that maximizes the quality of execution of an application, could be a bad configuration for another type of service. Moreover, different instances of the same application in the server can work in different conditions (e.g., different number of concepts template, different concepts of evaluation, different norms) and so they may require different configurations.

A. Evaluation list and risk factor

As introduced in a previous section, the execution of evaluation lists can be divided in two services that allow a specialist-role to either assess its own concepts template (from here on named risk assessments) or to share a single template with other specialist roles of the evaluation space service. The use of risk factor has the advantage of allocating possible risk values and guaranteeing static parameters to have a fair result to the norm´s concept evaluations. However, this solution suffers from high

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

resource consumption and context switching overhead, so it can be used in list spaces evaluations with a limited number of concurrent services. Therefore, when the number of users in an evaluation list space is high, the best solution is the use of concept templates whose execution is managed by an administrator-role by the LeSRi framework. Such role uses a LotusScript scheduling agent so the implementation of the specialist-role service has the duty of guaranteeing a fair evaluation list to the computational resources, for example, limiting the number of risk factors that an evaluation list can process in a single execution cycle. In some particular applications it is not possible to access services for all the actor-roles among the tasks of an actor´s space, so there are some actor-roles (e.g., specialist-roles) that should have priority accessing the computational resources of the evaluation list space.

B. Executive and Evaluation reports

In an evaluation list system where the computation is mainly based on the generation and processing of documents, the efficiency of the communication performance is a key parameter for the quality of an application. In LeSRi both local and remote communication can provide the evaluation list type report. In particular, current implementations of the software framework support the evaluation report on JAVA or PDF files. In an application, a large part of the concepts list is based on qualitative documents, the substandard conditions and reference parameters are also included, LeSRi allows replacing the traditional evaluation list for a single shared report with all other evaluation space specialist-roles.

C. Experimentation

The results of the different types of actor-role services execution can be analyzed by comparing the analysis of input information of the evaluated facilities. These examples involve two case types:

Manual, i.e., all the evaluations of the application space are services with transcribed text file.

Active, i.e., all the evaluation of application space where services that process all the documents received in each template of concepts is executed by the Specialist role.

Fig 3 shows the evaluation process time in hours.

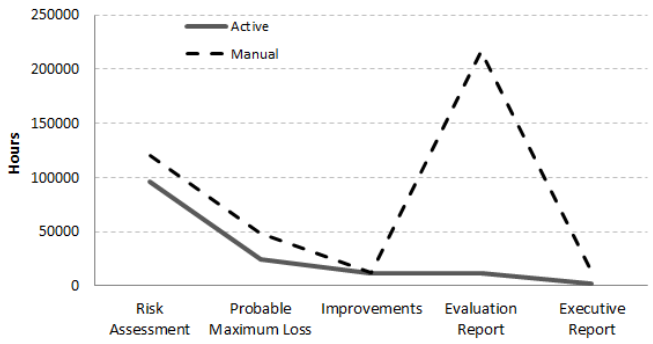

Fig.3. Evaluation process example performance.

The first service is based on the document exchange between evaluation list space tasks and concept generation from the template. The application starts with a specialist that creates a certain number of workplaces or facilities. The services in the event layer act upon the data received from the data sources, once the evaluation is done, data is sent to the final report [9]. Fig 4 shows the execution time of the application from 500 evaluation lists and the best evaluations are obtained from the facilities where the number of specialists increases.

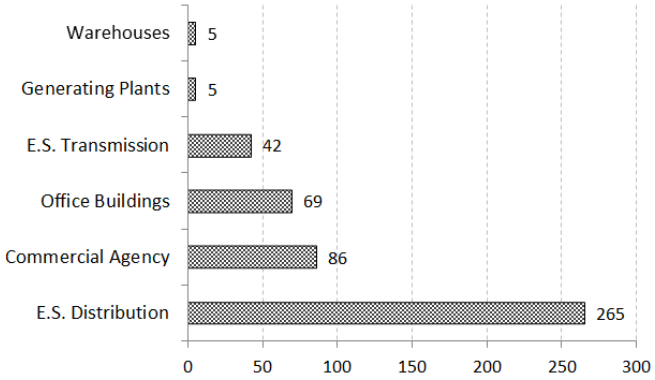

Fig. 4. Evaluated facilities example performance.

V. Conclusion

This paper presented a software framework called LeSRi that allows evaluating the risk level of facilities. The system uses concepts from international norms and assigned specialist-role services, combined to apply different implementations of the system tasks. The administrator-role can use other services to create concept templates for norms with hierarchically managed facilities.

LeSRi is implemented by using the LotusScript language in LotusDomino designer environment, it enables the rapid development and deployment of collaborative and workflow business applications [10].

LeSRi shares with UML [3], Skogan [4] and SOA [5], [6] the possibility to build applications that scale services architectures to a massive number of dynamic functions, but without the need of introducing new constructs that make the writing of services based programs complex. LeSRi has been designed for the evaluation of international standard norms with collaborative services, while the three previous software developments were designed for applications running inside multicode computers, the use of evaluation lists, risk factor and the maximum probable loss enables the implementation of complex interactions in a distributed application because a document contains all the qualitative and quantitative information to deliver a result and for creating and sending reports.

Current research activities are dedicated to extend the software framework to offer it as means for collaborative services and risks assessments, all in one system. Future research activities will be dedicated to the extension of the functionalities provided by the software framework and to

ISBN: 978-988-19253-6-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

its experimentation in different application fields. Regarding the extension of the software framework, current activities have the goal of providing a manual and active solution that obtains full advantage of facilities evaluation list features, enabling the specialist-role to generate reports with web services and calculate maximum probable loss. Additionally, future activities will be dedicated to the provision of a strategic management infrastructure to support the interaction between evaluation list spaces of different organizations. Current experimentation of the software framework is performed in the field of energy plants modernization and information integration, but in the future will be extended to the collaborative work services, mobile systems and off-line system services for the management of documental information.

ACKNOWLEDGMENT

Thanks to the engineers Roberto Butrón, René Ramirez and Claudio Flández for their cooperation allowing the scope to apply to all productive processes of the CFE.

Thanks to Martín Santos for his contribution with corporate portals knowledge.

Thanks to Israel Galvan for his advice in terms of risk assessment**.**

REFERENCES

[1] G. Agha, "Actors: A Model of Concurrent Computation in Distributed," *Cambridge, MA: MIT Press*, 1986.

[2] A. Poggi, "Developing Flexible Applications with Actors," *Advances in Information Science and Applications*, vol. 1, pp. 116-121, 2014.

[3] C. D. Ahmed Nait-sidi-moh, Jaafar Gaber y Maxime Wack, "Modeling and Specification of Web Services Composition Using UML-S," *International Conference on Next Generation Web Services Practices*, 2008.

[4] D. Skogan, R. Groenmo y I. Solheim, "Web service composition in UML," *Enterprise Distributed Object Computing Conference*, vol. Eighth IEEE International, pp. 47-57, 2004.

[5] T. Erl, Service Reusability and Service Autonomy, Prentice Hall, 2008.

[6] S. Hanna y M. Munro, "Fault-Based Web Services Testing," *Fifth International Conference on Information Technology: New Generations*, vol. Fifth International Conference, pp. 471-476, 2008.

[7] S. Gallardo Vera, E. Nava Lara y M. Santos Dominguez, "SISTEMA WEB PARA LA GESTIÓN DE APLICACIONES DE SEGURIDAD NORMATIVA," *IEEE - REUNION INTERNACIONAL DE VERANO DE PONENCIA RVP-AI GIN-10 PON 117*, vol. 32, p. 5, 2014.

[8] G. J. Orme y M. Venturini, "Property risk assessment for power plants: Methodology, validation and application," *Energy* , vol. 36, p. 3189e3203, 2011.

[9] Yemula Pradeep, P. Seshuraju, S.A. Khaparde y Rushikesh K. Joshi, "Flexible open architecture design for power system control centers," *Electrical Power and Energy Systems*, vol. 33, p. 976–982, 2011.

[10] I. D. Designer, "Lotus Domino," IBM, 2015. [En línea]. Available: http://www-03.ibm.com/software/products/en/ibmdominodesigner. [Último acceso: 6 July 2015].