A CLOUD BASED ARCHITECTURE FOR WORKING ON BIG DATA WITH WORKFLOW MANAGEMENT

Tahereh Koohi-Var[1] and Morteza Zahedi[2]

[1]International Campus of Kharazmi, Shahrood University of Technology, Shahrood, Iran
[2]CE Department, Shahrood University of Technology, Shahrood, Iran

ABSTRACT

In real environment there is a collection of many noisy and vague data, called Big Data. On the other hand, to work on the data middleware have been developed and is now very widely used. The challenge of working on Big Data is its processing and management. Here, integrated management system is required to provide a solution for integrating data from multiple sensors and maximize the target success. This is in situation that the system has constant time constrains for processing, and real-time decision-making processes. A reliable data fusion model must meet this requirement and steadily let the user monitor data stream. With widespread using of workflow interfaces, this requirement can be addressed. But, the work with Big Data is also challenging. We provide a multi-agent cloud-based architecture for a higher vision to solve this problem. This architecture provides the ability to Big Data Fusion using a workflow management interface. The proposed system is capable of self-repair in the presence of risks and its risk is low.

KEYWORDS

Workflow Management Interface, Multi-Sensor Data Fusion, Big Data, Cloud Computing

1. INTRODUCTION

Managers often reuse and refine workflows or current patterns of processes [1, 2]. They can share their workflows or create new workflows from a public repository or a new project space (the extracted data, new components, sub-workflows, etc). Meanwhile, the use and fusion of structured and unstructured datum that has a huge volume (Big Date [3]) and it seems to be a hard work. On the other hand, if the process is automated by workflows, data sources are selected during the run of workflow and the parameters are set by the user. Workflows may have resource scheduling in High Performance Computing (HPC [4]) (for example in local computer clusters), or time scheduling including remote resources (Grid computing [5] or Cloud [6]). Data may also be stepped i.e., from certain places waiting for data, where computational tasks are done on HPC clusters. For an instance of the application, it is important for marine vehicle to have tracking [7] and navigation [8]. This tracking and navigation is needed along with other critical time and inhibitor works in management. The nature of the inhibitor and critical time works has led marine vehicle management systems towards systems with "Multi-INT" multi sensor data fusion (Figure 1). But, generally the design of these systems has challenges, because gaining reliable information is difficult because of the complexity of the decision making in dynamic environments. In fact, the designer of management systems have equipped management systems with the ability of automated cognitive processes, on the importance of collecting various data from different sensors. This equipment is for increasing the security and efficiency of marine vehicles. In the equipped vehicles with the purpose of combining data and performing tasks as workflows is to bring the marine vehicle to a situation to increase its ability. Actually, our purpose in the management system of a marine vehicle is providing an interface for the tools that are presented as heterogeneous agents for safe navigation and mission execution. The system interfaces with various sensors to give real-time understanding of the situational maritime picture to the operator (the user). These sensors can include: GPS/DGPS, Radar (S&X band, and TWS) ,

Compass, weather, Electronic Support Measurement (ESM), Precision Direction Finder (PDF), RF system, Specific Emitter Identification (SEI) , Automatic Identification System (AIS).
The data set is supposed to help manage user-managed processes or controllers. In implementing a management system this data and other structured data are involved, and all these must be considered.

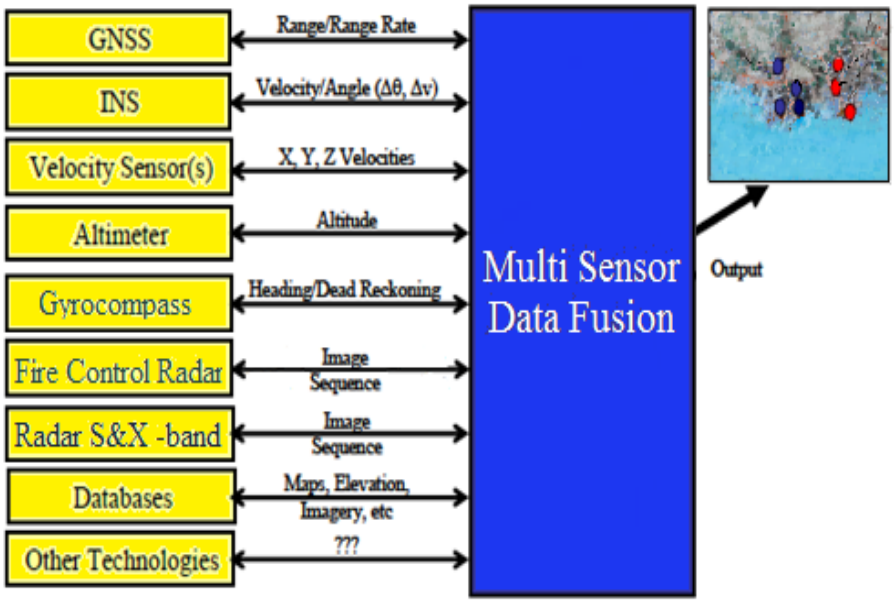

Figure 1 – Multi sensor data fusion of a marine vehicle

Since the marine vehicle has the reorganization ability based on a wide range of surface and sub-surface sensors, the data may have come from a variety of sources so that the aggregated data maybe vague and noisy. Here, the management system is required to provide a solution to the data defect and maximize the target's success. This system is needed to be able to manage processes, interpret and fuse the pieces of different types of information quickly. The dependency between the data fusion, the sensor set, the goals and the environment make the implementation of such a framework difficult. In marine environments, particularly, the large volume of work and the complex signal emission environment, especially for observing underwater sonar, increases the complexity of data fusion operations. The common feature of those systems that support data fusion is that they have several sensors that operate with a variety of data sources, have limited constant time for processes, and the decision processes are done real-time based on aggregated information. A reliable data fusion model should meet this requirement and allow the user to monitor data stream continuously. For this purpose, we propose a system for data fusion based on the multi-agent system model presented in [13], using the real-time operating system, and an Application Interface (API). This system must be capable to run the user-defined workflow automatically and can handle Big Data. For this purpose, a model for managing the workflows of this cloud-based system is presented that have the potential to manage risks. Finally, we evaluated the presented system by simulating it in the CloudSim framework [22].

The structure of this paper is as follows: In section 2 we discuss about the works related to issues, including issues related to middleware and workflow management system. Then in Section 3 we will introduce the proposed architecture provided for data fusion. In Section 4, the model for workflow management is provided based on cloud computing, and then we will provide a way to improve it. In Section 5 we show the results and simulation. Finally, in Section 6 the paper is concluded.

2. RELATED WORK

This study is concerned about the design of data fusion system, the middleware, the workflow management system, and the proposed architecture. In the next sub-sections, we study each of them.

2.1. DATA FUSION SYSTEM DESIGN ISSUES

Sloman [26] has a strong evidence for thinking and designing models at the level of architecture, instead of building independent component units. Designing the real world data fusion requires the collaboration of the distributed system (network), organizational concepts (command), and environmental perception (textures). In the design of a data fusion system for a particular composition task, the designer must choose from many selections are involved with trade-off. Lambert [16] identified eight issues should be considered by the designers of the data fusion systems within the scope of the Object assessment, Situation and Impact. The eight tradeoffs illustrated in [16] are as follows:

1. Automation- from totally manual to fully automatic, with all levels between.
2. Aggregation- including communication topology, protocols, roles, and bandwidth, etc.
3. Integration model - for the aspects of Human Machine Integration (HMI), Machine-Machine Integration (MMI), and Human-Human Integration (HMI)
4. Situational awareness - by human, machine, or both.
5. C2 Fusion Policy - centralized versus decentralized.
6. Representation - expressivity versus calculability.
7. Information processing - analytical versus heuristic.
8. Integration of the machine.

Each of these tradeoffs that are well-distinguished by Lambert, requires a branch of extensive research and expertise related to it. For example, consider the integration trade-off, which includes networking issues and communication protocols. In the navigation management with the security issue some protocols are required to be selected. The protocols may be capable to handle cryptographic issues as well as network configuration policies. They must be taken so that the datum is transmitted correctly. Meanwhile, a good data fusion system in the management is the system that can meet the system's control needs independent of these issues, without worrying about infrastructure technology. Therefore, in this paper, we will design an interface for workflow management according to the following:

1. Lightweight: This interface needs to be implemented on low-level platforms.
2. Multi-agent: Processes are done by a collection of processing agents [13]. The only requirement is that these systems are based on a real-time operating system [18], or a middleware is used that provides the processing of real-time processes. We'll explain this in more detail in the next section.
3. Portability: Data fusion processing platforms is not homogeneous (for example, they can be Personal Digital Assistants (PDAs), Laptops or sensor nodes). The data fusion interface is required to be not sensitive to the diversity of the operating systems and the architecture between the network nodes.
4. Development: Modular construction ensures to make programming easy.
    a. Scalability: The design requires scalability to be considered when increasing network size.

The migration or distribution environment of our proposed architecture contains a key component called middleware. In the reminder of this section we introduce the related works about this component.

2.2. MIDDLEWARE

The most common middleware used in management systems is the web services [17] and Common Object Request Broker Architecture (CORBA) [28]. But these have disadvantages. For example, unlike the success of CORBA in the deployment of the basic architecture of the distributed object systems, its learning is difficult, it is difficult to use, it suffers from several designs, it has low performance in the protocol, and it is necessary to support some of the features that always are required.

In this paper, we intend to consider our system based on multi-agent middleware, which benefits from the support of dynamic configuration qualities, displaying knowledge and understanding the relationships between elements involved. This middleware is required to provide different possibilities depending on the type of heterogeneous problems, for realizing communications between nodes which are distributed in a computational environment. Heterogeneity is often restrained by standard communication protocols that is expected be involved with all software, and technology network systems. To overcome the heterogeneity problems, we will provide a framework based on the current multi-agent platform. For this purpose, we will examine the current multi-agent platforms.

Current multi-agent platforms: JADE (Java Agent Development) is the most matured framework, after more than ten years of development and is the reference of multi-agent system development agrees with FIPA. Meanwhile, there are other related agent platforms that help developing these types of tools. The most related of these tools are summarized in Table 1.

Table 1: Current related agent-based frameworks

Related article	Name	Language	Standard	Free
[29]	JADE	Java	FIPA	Yes
[9]	Jadex	Java	FIPA	Yes
[10]	Cougaar	Java	No	Yes
[11]	Agent factory	Java	FIPA	Yes
[27]	JAMES II	Java	No	Yes
[12]	JACK	Java	No	No
[27]	AnyLogic	Java	No	No

Recently, a modern approach has been proposed for multi-agent development [27] that is comparable with JADE, and has features such as the ability to implement the basic management agent in C ++, Java and Python. This middleware is scalable, and it has many agents and short response time. So, it is suitable for the data fusion interface that we will provide.

2.3. ARCHITECTURE

Using a distributed system [25] allows the system processes to be managed at runtime. Among the previous work that has been done to manage processes at the operating system level we can refer to the work at [15]. The architecture presented in [15] is a distributed system. This system is based on the pattern of service process communication via message transactions. The communication protocol used in the system that proposed in [15] is TCP/IP and a system control and monitor processor selects data in HTTP. In [15] each fused data has a link to the data distribution system. The disadvantage of a distributed system over the multi-agent system is that the elements of a distributed system cannot have the autonomy of agents in multi-agent systems [13, 27]. Therefore, in this work, we propose a multi-agent based architecture that let the elements act as autonomous.

2.4. WORKFLOW MANAGEMENT SYSTEM

Workflows are rule-based management programs that they model the execution of tasks to the complex functions and direct the work [19]. Workflows are usually used to automate the hidden layers from the user's perspective. Middleware like CORBA can also benefit from the workflows for doing their work [19]. But for workflow management it is necessary to consider the volume of data has become big, and so the management should be capable to work well with Big Data [4], [6]. On the one hand, this system should be responsive to the big volume of the data and, on the other hand, it should have low cost. The issue of lowering the energy costs has been investigated in different ways in the system [24]. We therefore propose a method based on the cloud computing that its cost is low as the services are provided over time.

3. ARCHITECTURE

The proposed architecture is constructed of radar, tracking fusion and workflow monitoring and control agents. This architecture is a similar the architecture proposed in [15], with the difference that in this architecture the multi-agent middleware is used. In this architecture, the user interface and mission goals must be defined in the system control and monitoring agent. Figure 3 illustrates the architecture proposed in this study. In this architecture, a multi-agent middleware is considered, which is connected to 2 weeks and 24 hours storages. On the other hand, the middleware is connected to acoustic sensor agent, data fusion agent, server, receiver and the information agent. Each of these can work autonomously. Figure 2 illustrates the proposed middleware based on [13]. The required middleware has a multi-agent class diagram.

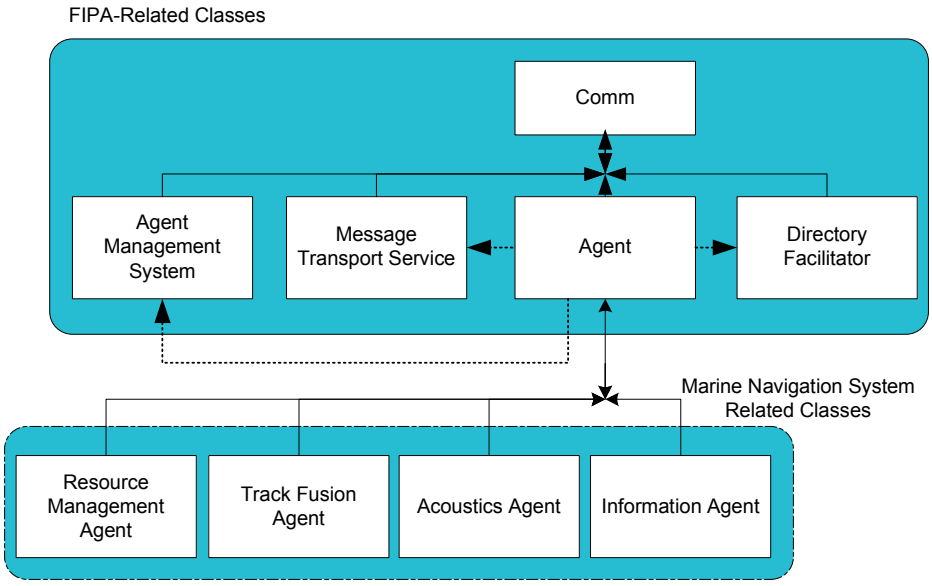

Figure 2- Proposed Multi-agent class Diagram (Middleware)

The used communication protocol in the proposed architecture (Figure 3) is based on FIPA. This protocol is usually used in the multi-agent systems and facilitated the development of the agent-based applications [13].

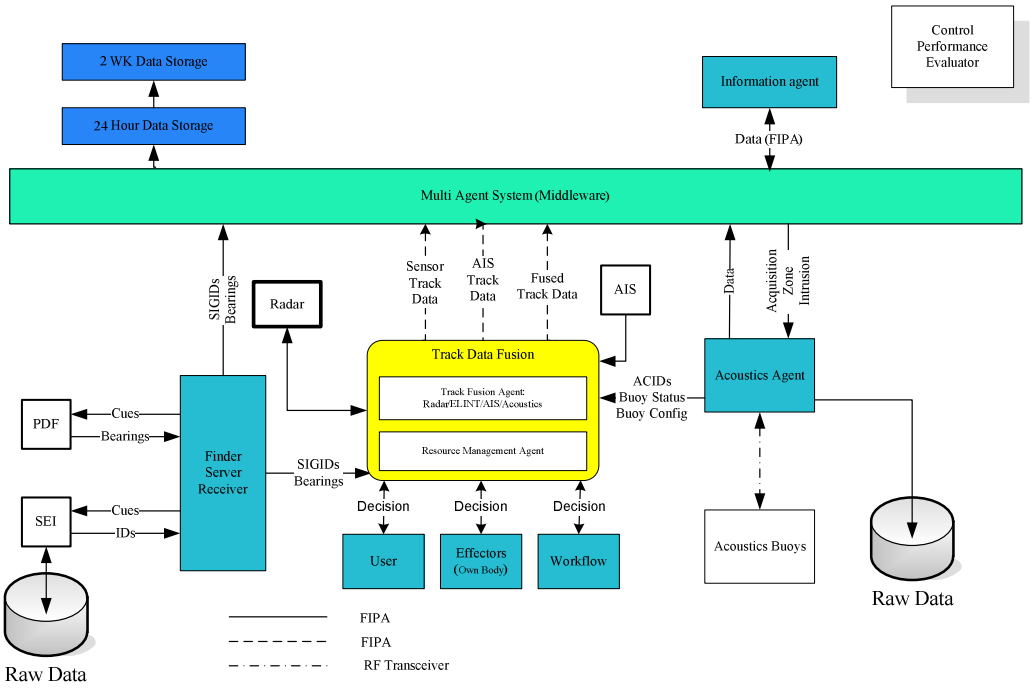

Figure 3 – Proposed architecture schematic

The information agent is designed to monitor databases, raw sensor data filtering, topology management, and so on. This agent takes the observations of the real world system. During the execution of the workflow by the agent, the input datum has been consumed and new data products have been created (in the track data fusion agent). In this architecture for implementing large-scale performance (the workflow is played on hundreds or thousands of nodes: for hours, days, or weeks at a time), real-time monitoring is also important: mid-data products, and specific information about web-based monitoring is required to inform the user about the progress and possible issues during execution.

Further details of this architecture are as follows:

Sensor Track Data Fusion: Figure 4 shows our proposed architecture. Sensor data (e.g. ELINT/Radar, etc.) will be fused in the data fusion agent. In fact, the input is a real-world observation to estimate the real world state. A weather prediction system, which is covered as an agent, will be into this category. The output of this agent is based on the National Marine Electron Standards (NMEA).

Track fusion agent: Modelling the navigational dynamics of the marine vehicle leads to 6 Degrees Of Freedom (6DOF), so 3 orthogonal angle rates and 3 orthogonal accelerations are required.
Resource Management Agent (RMA): In this architecture, we separate information fusion functions and Resource Management (RM). This separation was the goal of the promotion from the JDL models to DFIG [14]. RM is a set of management tools, including the management sensors, workflows and manoeuvre, platform placement and the user selection for meeting the mission goals. Since the unobserved aspects of situation awareness problem cannot be processed by a computer, the user's knowledge, and reasoning are essential; in fact, to provide and maintain the Quality of Service (QoS), RM must be guided by QoS. The workflow management is done by this agent.

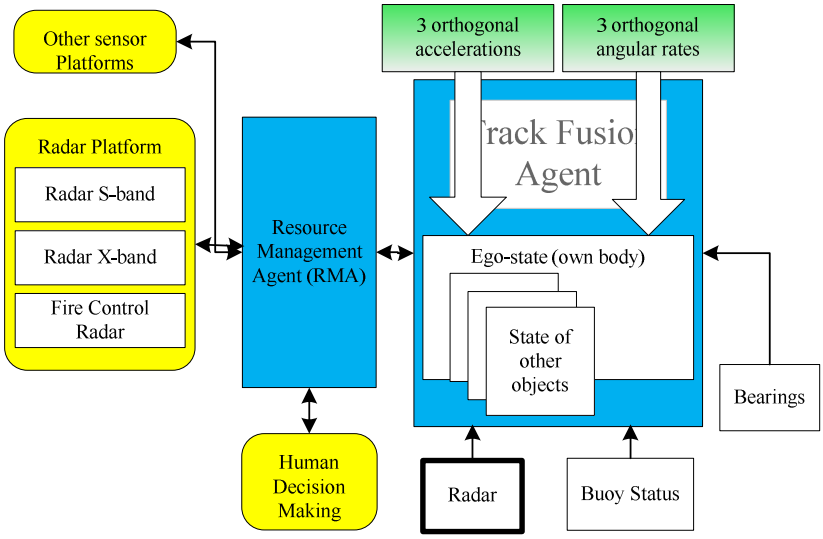

Figure 4. Fusion Architecture core Schematic

Figure 5 shows that at the core of the proposed architecture the RMA agent is consisted from the manoeuvre management, sensors and workflow agents.

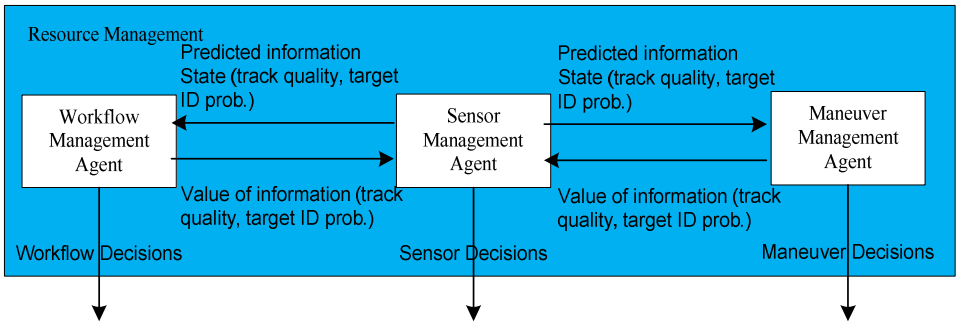

Figure 5. Interface between sensor management, workflow, and manoeuvre agents

RMA is an estimate of the current state of the global situation and a plan or a produced solution. Such agent can use other agents to for example find additional information. In the next section, we explain the workflow management model to implement by this agent.

4. WORKFLOW MANAGEMENT MODEL BASED ON CLOUD COMPUTING

To work with Big Data, a cloud-based management model is selected for workflow management. With the transfer of the data in the cloud, the cloud computing has the challenge of security and the security measure is unknown [21]. The distance between the user and the physical location of the data creates a barrier called risk. Risk is the possibility of accident happening and has an unwanted influence on performance goals. Different types of security risks, integration, accessibility, and efficiency are raised. The risk characteristics of the impact, likelihood, and effects for categories of the types of strategic, technical, operational, and rule based. These are the bases of the work: the risk management problem formulation, risk modelling, risks scoring, flexible risk system modelling, and finding control point of the cloud computing, modelling to achieve decision making response, and providing the optimal solution for decision making. In this regard proposing a model for information risks assessment is necessary to assess the allocation

and maintenance of resources. The following methodology is our proposed approach for evaluating the efficiency of workflow management system in the presence of risk.

System performance improvement in the presence of risk: the proposed approach to risk management in workflows is based on the work presented in [23]. We initially estimate the system failure rate based on the Poisson distribution like [24]. Then, we must balance the systems loads with the migration of workflows in the proposed framework. In fact, migration is a basic concept of the mechanism of proposed workflow management method. Thus, we give the cloud system the capability of self repair to have more safety against risks. In the next section, we evaluate our simulations regarding the proposed framework.

5. SIMULATION AND EVALUATION

For simulation we have used Cloudsim [22]. First, we proposed a cloud platform based on the characteristics of Table 2 and we considered having per physical machine a virtual machine. In fact, each host in the cloud is dedicated to a virtual machine. Then, we calculated the system risks based on the failure rate that may be caused by the attacks on the cloud.

Table 2: Cloud specifications

Virtual machine	operating system	Processor	RAM	Workflows
50	Linux	4 cores	4	2

In this model, the failure rates are applied randomly with a Poisson rate to a range of workflow (Cloudlet). In some runs, we may not have failures. But, the system in the case of failure can repair itself by reallocating the virtual machine list to the system hosts.

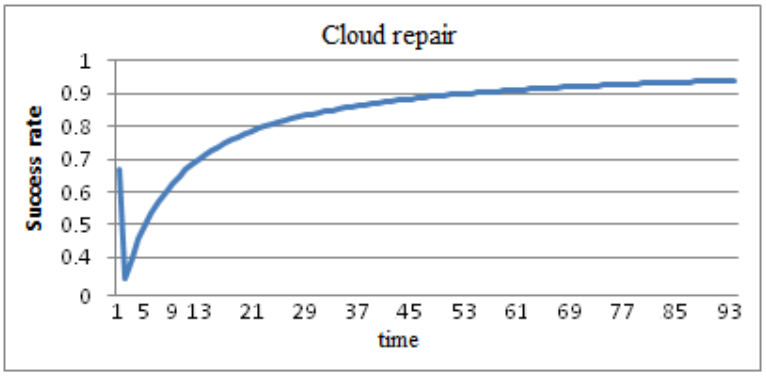

Figure 6. Cloud repair in the workflow management system-vertical axis: success rate, and the horizontal axis: Time

Figure 6 shows the system ability in self-repair. It shows that the success rate is increased over time that it shows the system has the ability of self-repair in the case of failure. Figure 7 shows the availability rate of the system in the same conditions.

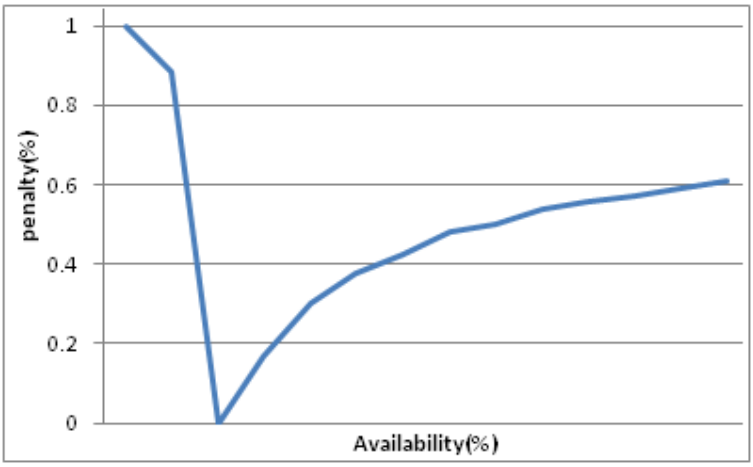

Figure 7. Workflow management system availability to the cloud penalty- vertical axis: penalty, and the horizontal axis: availability

Figure 7 shows the system has the ability of self-repair. The results form Figures 6 and 7 show that the cloud risk is low.

6. CONCLUSION

In this paper, we presented an interface for workflow management to monitor the performance of a marine, and then we stated its challenges. We then presented an architecture that can handle the process management needs in the case of navigation and tracking, according to today technology. This architecture can meet Big Data requirements in implementation. On the one hand, because it is a cloud-based workflow management model and on the other hand the results show that it can work well in the presence of risk. Migration has the main role in this model in the self-repair and the system control in the presence of risk.

ACKNOWLEDGEMENTS

The authors wish to thank their colleagues at Web Mining and Pattern Recognition (WMPR) Lab of Shahrood University of Technology for their participation, insights, and patience.

REFERENCES

[1] T. Koohi, and M. Zahedi, Scientific Workflow Clustering based On Motif Discovery, International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), 7 (4), 2017.

[2] T. Koohi, and M. Zahedi, Linear Merging Reduction: A Workflow Diagram Simplification Method, 8th International Conference on Information and Knowledge Technology, 2016.

[3] F. Bajaber, R. Elshawi, O. Batarfi, A. Altalhi, A. Barnawi, Sh. Sacr, Big Data 2.0 Processing Systems: Taxonomy and Open Challenges, Journal of Grid Computing, 14 (3): 379-405, 2016.

[4] E. Deelman, T. Peterka, I. Altintas, Ch.d. Carothers, KK van Dam, K. Moreland, M. Parashar, L. Ramakrishnan, M. Taufer, and J. Vetter, The Future of Scientific Workflows, International Journal of High Performance Computing Applications, p. 1-17, 2017.

[5] AG Finogeev, DS Parygin, AA Finogeev, The convergence computing model for large sensor data mining and knowledge discovery, Human-centric Computing and Information Sciences, 7 (1): 11, 2017.

[6] P. Kacsuk, J. Kovacs, Z. Farkas, The Flowbster Cloud-Oriented Workflow System to Process Large Scientific Data Sets, Journal of Grid Computing, 1-29, 2018.

[7] T. Koohi, F.K. Dehkordi, A. Izadi-Pour, and M.N. Fesharaki, Optimization of Multi-Target Tracking in a Multi-Agent Architecture with Multi-Sensor Data Fusion, Majlesi Journal of Electrical Engineering, 4 (1): 2015.

[8] T. Koohi, M. Yaghoobi, and S. Toosizadeh, Advances in Fuzzy Path Following of a WMR to Guarantee Successful Navigation, 5th International symposium of SASTech, 2011.

[9] A. Pokahr, L. Braubach, W. Lamersdorf, Jadex: A BDI reasoning engine. Multiagent Systems Artificial Societies and Simulated Organizations, Springer, 15: 149-174, 2005.

[10] A. Helsinger, T. Wright, Cougaar: A robust configurable multi agent platform. IEEE Aerospace Conference, 1-10, 2005.

[11] R. Ross, R. Collier, GMP OHare, AF-APL-bridging principles and practice in agent-oriented languages. In: Second International Workshop on Programming Multi-Agent Systems (ProMAS'04). Lecture Notes in Computer Science, Springer, 3346: 66-88, 2004.

[12] M. Winikoff, JACK TM intelligent agents: an industrial strength platform. Multiagent Systems, Artificial Societies, and Simulated Organizations, Springer, 15: 175-193, 2006.

[13] D. Vallejo, J. Albusac , JA Mateos , C. Glez-Morcillo, L. Jimenez, A Modern Approach to Multi-Agent Development, The Journal of Systems and Software 83: 467-484, 2010.

[14] E. Blasch and S. Plano, DFIG Level 5 (User Refinement) issues, Supporting Situational Awareness Reasoning, In Proceedings of the International Conference on Information Fusion 2005.

[15] DJ Bielecki, NRL Global Vessel Tracking Project (VTP), Naval Research Laboratory, 2008.

[16] DA Lambert, Tradeoffs in the design of higher-level fusion systems, Proceedings of the International Conference on Information Fusion, 2007.

[17] D. Booth, H. Haas et al. (Eds.) Web Services Architecture, W3C Working Draft, 11 February 2004. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

[18] KH Kim, C. Subbaraman , Principles of Constructing a Timeliness-Guaranteed Kernel and Time-Triggered Message-Triggered Object Support Mechanisms, ISORC, pp. 80-89, 1998.

[19] F. Ranno, and SK Shrivastava, A review of distributed workflow management systems, in the Proceedings of the international joint conference on Work activities coordination and cooperation, 1999.

[20] R. Wu, Y. Chen, E. Blasch, B. Liu, G. Chen, D. Shen, A container-based elastic cloud architecture for real-time Full-Motion Video (FMV) target tracking, In Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1-8th of April 2014.

[21] M. D. Ryan, Cloud Computing Security: The Scientific Challenge, and a Survey of Solutions, Journal of Systems and Software, 2013.

[22] RN Calheiros, R. Ranjan, A. Beloglazov, CA De Rose, R. Buyya, CloudSim: a toolkit for modeling and simulation of cloud computing environments and assessment of resource provisioning algorithms. Software: Practice and experience, 41 (1): 23-50, 2011.

[23] JO Fitó, J. Guitart, Business-driven management of infrastructure-level risks in Cloud providers, Future Generation computer systems, 32: 41-53, 2014.

[24] K. Djemame, D. Armstrong, J. Guitart, M. Macias, A Risk Assessment Framework for Cloud Computing, IEEE Transactions on Cloud Computing, 4 (3): 265-78, 2016.

[25] H. Pérez , JJ Gutiérrez , S. Peiró, A. Crespo, Distributed architecture for developing mixed-criticality systems in multi-core platforms, Journal of Systems and Software, 123: 145-59, 2017.

[26] A. Sloman, What sort of architecture is required for a human-like agent? In Foundations of Rational Agency, ed. M. Wooldridge, and A. Rao. Dordrecht, Netherlands: Kluwer Academic Publishers, 1999.

[27] K. Kravari, N. Bassiliades, A survey of agent platforms, Journal of Artificial Societies and Social Simulation, 18 (1): 11, 2015.

[28] W. Henrich, Th. Kausch F. Opitz, Data Fusion for the Finnish Fast Attack Craft Squadron 2000: Concept and Architecture, In the Proceedings of the International Conference on Information Fusion, 2004.

[29] FL Bellifemine, G. Caire, A. Poggi, G. Rimassa, JADE: A software framework for developing multi-agent applications. Lessons Learned, Information and Software Technology, Elsevier 5 (1-2), 10-21, 2008.

AUTHORS

Tahereh Koohi-Var is a PhD student at WMPR Lab of Shahrood University of Technology with Master of Science from Azad University of Mashhad (2012). She obtained Bachelor Degree in Computer Engineering from Azad University of Mashhad in 2009. Her researches are in fields of data fusion, signal processing, and business process management systems. Recently, she has focused on BPMSs. She is currently CEO at enahang Co. Ltd.

Morteza Zahedi is an Assistant Professor at Shahrood University of Technology. He is currently CEO at Ganjineh Co
.