SindhUniv. Res. Jour. (Sci. Ser.) Vol. 51 (2) 359-362 (2019)
http://doi.org/10.26692/sujo/2019.6.58
SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Evaluating VOML Framework as a Dependable System

NOOR JEHAN RAJPER

Institute of Mathematics and Computer Science, University of Sindh, Pakistan

Received 14th September 2018 and Revised 11th April 2019

Abstract: Virtual Organizations (VOs) and Virtual Breeding Environments (VBEs) are resilient, dependable and fault-tolerant by virtue of being virtual in existence. In this article, Virtual Organization Modeling Language framework is evaluated against dependability and fault-tolerance aspects.

1. INTRODUCTION

The major feature that makes VO a promising solution to unique challenges of current unpredictable and highly volatile and other information systems is that it could be formed on-demand out of autonomous (and perhaps competing) entities that might complement each other for this particular business opportunity that has arisen, is, it can readily be relied upon. This reliance is recognized as dependability attribute of any system (Silva *et. al.*, (2018), (Mahato *et. al.*, (*2018*) (Wears, *et. al.*, 2018) .A system can exhibit dependability at many different levels such as fault-tolerance and dynamic adaptation in either its logic or composition.

VOML is a framework that offers three different modeling languages each catering for different aspect of VO description. Since this framework is specifying VOs, it must support dependability through its different constructs. In this article, VOML framework is evaluated against various flavors of dependability. Due to the shortage of space this article does not explain in details the VOML framework (interested readers may refer to (Rajper, (2012)), relevant details of the VOML framework are provided where necessary to understand its approach towards offering particular feature of dependability.

Related Work

Generic Modelling frameworks such as UML, ERP, Petri Nets have been used to model VOs. For example, in (Bryans, *et. al.*, 2006) Vienna Development Method is used to model a VO for the purpose of carrying out analysis and verification. However, unlike VOML, it does not focus of dependability aspect of the modeled systems specifically. On the other hand (Norman, *et. al.*, (2004))like VOML, allows tasks to be shared, however they only allow sharing of a task based on capacity whereas VOML allows sharing based on capabilities as well. The field of dynamic adaptability in general (Serugendo, *et. al.*, 2007) is relevant. ASSL (Vassev, *et. al.*, 2009) provides a modeling language for autonomous systems that in particular focuses on flexibility in dynamic adaptation. ASSL abstracts away from level requirements of the system, however these are at the heart of VO modelling. Policy languages are one of the widely accepted ways of reconfiguring systems to make them more dependable (Vassev, *et. al.*, 2009)). Often policy languages describe low level system management aspects (Lupu, and. Sloman, 1999); at higher levels of abstraction the APPEL (Blair, *et. al.*,(2005) policy language has been adapted for work flows, where it allows to insert and delete tasks (Gorton, *et. al.*, 2009). For VO's no such language currently exists but the presented reconfiguration language is a variant of APPEL.

Dependability Evaluation Dimensions

In order to evaluate the dependability aspect of VOML, this research work looks at dependability from two different dimensions:

Business commitment Dimension:

This dimension refers to the ability of a system to recover from events hindering the fulfilment of the business workflow requires a courier service then a non-dependable business will fail their whole business commitment if that specific courier service provider fails to fulfil its commitment whereas, a dependable system will look for another courier provider instead of halting the workflow seamlessly.

.

a) **Internally Induced Failures:** These kinds of failures occur due to local failures of a VO such as disputes suddenly erupt between VO member or a member leaves the VO without honoring its part of the commitment.

b) **Externally Induced Failures:** These kinds of failure are reaction to the changes in the environment external to the VO such as change in government policies forbids business partnership with members from certain countries as a result those VOs which already offer some service in collaboration with any member from those countries must have to undergo changes and that is only possible the VO exhibit this flexibility.

1. Infrastructural Dimension:

Problems in communication links, such as broken communication lines or delays in communication lines might fail VOs from accomplishing the VO goal when the circumstances of business environment has changed opportunity has already been lost.

a) **Internally Induced Failure:** An example of internally induced failure can be when the connection to a VBE provided Database (to a VO) is suddenly broken; as a result the operation of VO comes to halt.

b) Externally Induced Failures: An example such failure is when a VBE uses public or private networks not owned by VBE, to let its globally scattered participants interact with each and that network encounters any problem.

VOML Evaluation Using Commonly Occurring Failure Scenarios

VOscope with failures by being resilient, such as by seeking complementary business partners with when one organization fails to achieve a business opportunity, reserving more resources (physical, human skills, etcetera)in order to expedite the progress, evolving structurally and operationally aligning it self to changing business scenarios, etc. Yet there can be situations when eventually VOs might fail. This section is going to evaluate how the VOML framework handles failures and how dependable this framework is? Given below are most frequent causes of the failure:

1. ICT infrastructure failure: The VOML framework at the moment does not does not cater for infrastructure related issues and it is the responsibility of the VBE-enabling infrastructure to cater for those aspects. However, VOML framework is flexible enough to describe policies specifying alternatives to infrastructure related problems such as re-route the communication to a different channel if the current link is congested or down.

```
Policy Comm Channel Degraded applies
To ICT channel
When Below Thres hold(commChannel X,
threshold Value)
do
switch Channel(commChanelY)
```

Above policy (*Comm Channel Degraded*) is activated by the triggering of `Below Throshold` event which informs that the communication channel named *comm. Channel X* has fallen below threshold value mentioned in the *threshold Value* parameter. This policy in turn changes the communication channel to *commChannel Y* as a result by executing the `switch channel()` action.

2. Fixed organizational structures: Fixed organizational structure can lead to internally induced failures of infrastructure dimension. A VO has to keep reconfiguring itself to such that its structure accommodates the changing business environment. An operational derived out of that structural model is limited in its ability to adapt as it is technology dependent. The VOs requires far more flexibility then dynamic discovery and binding of entities at run-time. The VOML framework has answered this problem by keeping the structural aspect of the VO separate from the operational aspect. The structural aspect is abstract enough to allow for far more flexibility in changing different aspects of VO description, out of which operational model is derived using mapping rules (beyond the scope of this article). This allows VOs to be far more flexible to change different aspects of the VO description at different levels of abstraction leading to more dependable VOs.

3. *Conclusion:* VOML thorough its reconfiguration policies and keeping abstract and technology closer concrete details separate allows for the VBEs and VOs to be fault-tolerant and dependable systems for both externally induced as well and internally induced failures of infrastructure dimension.

4. <u>CONTRACTVIOLATION:</u>

Contract violation is regarded as failure to oblige its commitment and this is an example of internally induced failure of business dimension. For example, whena VO member fails to provide the promised service or provides in less quantity then actually promised, by offering poor quality of service, by failing to provide on time, etcetera. The VOML framework allows to, define how to handle such violations VO-R modeling language. Policies can be defined using VO-R specifying the repercussions of contract violations.

Conclusion: The VOML framework can specify the situations through VO-R policies both at VBE level and VO level that describe how to recover from contract violations. The contract violations are not captured through VO-S and VO-Olevels though and theirre solution is at the discretion of VBE and VO management as well.

5. **Unforeseen external event:** A VO is bound to encounter externally induced business commitment failures during its lifetime. The VOML framework does not offer provisions to explicitly specify such situations but it can handle most of the failures seamlessly using VO-R policies. An example is when a transient member (a transient member is one hired outside VBE members) of a VO supplying one of the crucial catalyst ceases to operate suddenly and the VO has to look for new members and until it is decided that the new cataly stisfit for the chemistry the member providing that catalyst is included in the VO as temporary and once the catalyst has been evaluated to be suitable replacement, only then the catalyst provider's membership is changed to permanent one by making it associate of the VO. But for this to happen, the policy describing the event and what to do in the after math, must be described in advance.

Conclusion: While the VOML framework is rich enough to deal with many sort of externally induced business commitment failures, it is the vision and sharpness of management people in the business domain 1.
that matters most. The more visionary the management 2.
is the more unfore seen situations can be described in advance and the more a dependable the VO becomes.

Conclusion and Future work

In this article the VOML framework has been evaluated against dependability criteria to show how reliable this framework is in the face of unpredictable incidents. For the purpose dimensions of dependability are set and then VOML framework was checked against those dimensions. Overall, it is concluded that provision of reconfiguration language is flexible enough to cater for any kind of dependability aspects however at present VOML does not explicitly provide events, conditions and actions that cater for Infrastructure dimension. Therefore, in future this direction will be pursued.

REFERENCES:

Bryans, J., Fitzgerald, J., Jones, C., Mozolevsky, (2006) Formal modelling of dynamic coalitions, with an application in chemical engineering. In: Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation, 91–98

Blair, L., S. Reiff-Marganiec, K.. Turner, (2005) APPEL: The ACCENT Policy Environment/Language. Technical report CSM-164. Department of Mathematics and Computer Science. University of Stirling.

Di MarzoSerugendo, G.., J. Fitzgerald, A. Romanovsky, N. Guelfi, (2007) A metadatabased architectural model for dynamically resilient systems. In: SAC 2007 (ACM Symposium on Applied Computing), pp. 566–572. ACM, New York

Gorton, S., C. Montangero, S. Reiff-Marganiec, L. Semini, (2009) :StPowla: SOA, policies and workflows. In: Nitto, E.D., Ripeanu, M. (eds.) ICSOC 2007. LNCS, vol. 4907, 351–362. Springer, Heidelberg Safas

In: Zamojski W., J. Mazurkiewicz J. Sugier T. Walkowiak J. Kacprzyk (eds) Complex Systems and Dependability. Advances in Intelligent and Soft Computing, vol 170. Springer, Berlin, Heidelberg

Lupu, E. M. Sloman, (1999) Conflicts in policy-based distributed systems management. IEEE Transactions on Software Engineering 25(6), 852–869

Mahato D. P. and R. S. Singh, (*2018*) "Reliability Modeling and Analysis for Deadline-Constrained Grid Service,"32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), Krakow, 75-81.

Norman, T. J., A. Preece, N. R. Jennings, M. Luck, V.D Dang, T.D Nguyen, V. Deora, J. Shao, W.A Gray, N.J. Fiddian, (2004) Agent-based formation of virtual organisations. Knowledge Based Systems Vol. 17, Issues 2–4,103-111, Sciencedirect

Rajper N. J. (2012) VOML: Virtual Organization Modelling Language, PhD Thesis https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.555483

Silva E. F, D. C. Muchaluat-Saade N. C. Fernandes (2018) ACROSS: A generic framework for attribute-based access control with distributed policies for virtual organizations, Future Generation Computer Systems, Vol. 78, Part 1 1-17,

Toporkov V., A. Tselishchev D. Yemelyanov A. Bobchenkov (2013) Dependable Strategies for Job-Flows Dispatching and Scheduling in Virtual Organizations of Distributed Computing Environments.

Vassev, E.., M. Hinchey, (2009) ASSL: A software engineering approach to autonomic computing. In IEEE Computer Vol. 42(6), 90–93

Wears, R. L.., K. H. Roberts, (2018), Special issue, Safety Science, High reliability organizations and resilience engineering, Safety Science