



Assessing VOML Framework in the light of Modelling Language Criteria for Virtual Organizations

NOOR JEHAN RAJPER

Institute of Mathematics and Computer Science, University of Sindh, Jamshoro.

Received 13th July 2014 and Revised 18th November 2015

ABSTRACT: Virtual Organization Modelling Language (VOML) is a formal framework for specifying Virtual Organizations (VOs) and their Virtual Breeding Environments (VBEs). It consists of three sublanguages: (a) VO-Structural language (VO-S for short), that specifies structural aspects of VOs and VBEs, (2) VO-Operational (VO-O) language specifies operational aspect of VOs out of VO-S descriptions that are closer to underlying execution environment, (3) VO-Reconfiguration (VO-R) language allows both of these models to be reconfigured such that the models can be adapted and evolved to the changing business environment. In this paper the VOML framework is assessed for its suitability for modelling different requirements specific to VOs domain. The framework is assessed against criteria defined for modelling language of VOs as laid out in prequel and against ARCON reference model.

Keywords: Virtual Organizations, Modelling languages

1. INTRODUCTION

VOs come to play their role in hostile, competitive and disruptive environments by evolving and adapting to the changing environment (Hale *et al.* (2014), Cummings, *et al.* (2008), (Camarinha-Matos, *et al.* (2005), (Foster, (2001). This requires a sound understanding of their behaviour and unambiguous description of their models that aids the capability to reason about their core aspects. Hence, a modelling language that formally models VOs is indispensable (Sun *et al.*, (201), (Camarinha-Matos, *et al.*, (2008), (Rajper 2014). VBE defines a base long-term cooperation agreement among its participants (institutions or individuals) and characterizes their interoperable infrastructure. As such, a VBE represents the organisational context in which the creation and operation of VOs takes place; VOs are seen as ensembles that are formed dynamically to provide high-level functionalities, or services, by sharing a number of resources in a distributed way.

Evaluating and analysing different aspects of constantly changing VOs by specifying them first in a suitable modelling language becomes of paramount importance. Hence, it becomes very important that the modelling language chosen for modelling VOs offers constructs that can easily and readily specify peculiar characteristics of VOs and VBEs.

This paper, evaluates the VOML framework through the lens of criteria set out in the prequel of the paper.

The VOML framework is a compendium of three modelling languages; each targets a particular aspect of VOs and VBEs. The VO-S focuses on structural aspects and many of the characteristics peculiar to VOs such as relationship between two members, etc. The VO-R on the other hand permits different reconfigurations on the structure of the VO. These reconfigurations change the core model itself. The VO-O drives operational models of VOs in more details, out of VO-S model.

The rest of the paper is organized as follows. Section 2 provides a brief synopsis of the VOML framework relevant for evaluation done in Section 3. Thorough details of the framework can be found in (Rajper, (2012). Section 3 comprises of comparing the framework with the criteria designed for evaluating a modeling languages for VOs and VBEs. In Section 4, the framework is compared with ARCON (Camarinha-Matos, 2008) reference model. Finally, Section 5 presents some conclusion and future work.

2. MATERIALS AND METHODS

There is no universally accepted yardstick that can be used to weight any modelling language for its suitability to capture defining features and characteristics of VOs and VBEs.

In the prequel of the paper, an attempt has been made towards developing the criteria against which different modelling languages can be weighted. In this paper, the criteria are revisited in order to evaluate the suitability of VOML framework which consists of three

⁺⁺ Corresponding Author email: noor_rajper@hotmail.com

sublanguages, each covering different aspects of VOs and VBEs.

2.1 VOML FRAMEWORK

There are three different levels of representation in VOML. Level (1) defines the persistent functionalities of the VBE; level (2) defines the “*business configuration*” of a VBE at a specific moment in time in terms of VOs currently operating in the VBE; and level (3) represents the ensemble of components (instances) and connectors that, at that time, deliver the services offered by the VOs present in the business configuration, what is called a “*state configuration*”. The VO-S incorporates domain terminology as language constructs to define structural aspects of VOs at business configuration level as well as fixed organization structure of VBE at level (1). VO-R caters for the adaptability needs of VOs at the business configuration as well as at the state configuration level. The state configuration level focuses on the business functionality offered by any VO, in sufficient detail to allow for ready execution.

3. ASSESSMENT OF FRAMEWORK ACCORDING TO CRITERIA

In the prequel of the framework the author had proposed preliminary criteria (Rajper N.J., (2014)) to evaluate the suitability of a modelling language for VBEs and VOs. This paper evaluates each criterion with relevant aspects of the framework. It is worth mentioning here that the different criteria are visited that suits the flow of the paper which has lead to the random ordering of criteria with respect to (Rajper N.J., (2014)).

Criterion C1: *Representing domain concepts as modelling language constructs.*

The criterion C1 is about expressiveness of the language in terms of domain concepts. The readability and writability of a language is greatly enhanced if the stakeholders of the domain are able to specify a system in terms of the terminology they are most familiar with. Besides, it also expedites the learning and wide acceptance of the language in the targeted domain

Evaluation: The VOML framework satisfies this criterion by using domain specific terminology at each level; in fact VOML goes a step further by incorporating and aligning domain concepts with specific level of technical details exposed at that level. For example, at first and second level (persistent and business configuration levels respectively) domain concepts are used as “first class” entities in the language constructs by the sublanguages developed for that levels i.e VO-S and VO-R languages. The VO-S language defines structural aspects of the VBEs and

VOs using constructs such as Members (Partner, Associate and ExtEntity), Process, Data-Flow, Task (AtomicTask, ReplicableTask, and ComposableTask)etc. The VO-R language adapts the current configuration of a VO by applying reconfiguration policies to it. The set of different events, conditions and actions also reflect the domain concepts.

The third level (state configuration) which is more oriented towards underlying technical architecture on which the VO is going to eventually execute (such as Agent-technology or Service-oriented architecture) incorporates constructs that make more sense for IT community. The VO-Operational (VO-O) defines the instances of VOs currently operational in a VBE hence, the VO-O terminology provides terms such as components, wires (for coordination) and interfaces. Enthusiastic reader can refer to (Rajper 2014) for detailed description.

Criterion C2: Ability of a modelling language to evolve the VO model.

The Criterion C2 highlights one of the most crucial characteristics of languages of contemporary era that is riddled with unpredictability. The environments in which systems operate keep changing all the time. Therefore, a modelling language must offer means to modify the model so that it does not get obsolete and cope with changing circumstance. Many policy languages target this reconfiguration aspect specifically. Evaluation: The VOML framework caters for this criterion by VO-R sublanguage. The unique attribute of VO-R is that it targets two different levels of abstractions of a system: (1) at structural level where it is relatively easy to modify any aspect of the model due to the higher level of abstractions provided at that level. (2) it also offers the ability to adapt more concrete level models of the VOs that are closer to execution environment in which VOs operate.

Besides provisions for adaptations at two different levels, the framework also allows for modifications done at structural level update the operational model as well to reflect the changes made at the structural level. The criterion C2 is shown through different policies that reconfigure a VO model either at structural level or at operational level throughout the paper.

Criterion C3: Represent changing association of members with VBEs and VOs.

In VBEs some participants are permanent members of VBE and some are invited in for a certain period while their capabilities and skills are required by the VBE. Once they have fulfilled their assigned tasks they might leave the VBE or the VBE might invite them to

be permanent members of it or vice versa; in which case their association with VBE change from being temporary member to permanent member.

Evaluation: VOML support three different types of participants for VBEs and VOs: (a) Partner, who is a permanent member, (b) Associate, transient member of VBE and (c) ExtEntity, who is discovered at runtime only for a specific instance of VO. Furthermore, a member's type could be changed from one to another using relevant policies of VO-R. Given below is a self-explanatory example of such policy specification.

```

policy membersRoleUpdate appliesTo
CatalystProvision
when MemberLeft(CP)
do changeRole (CatalystProvision,
ExtEntity)...
policy newCatalystSourceSuitable appliesTo
CatalystProvision
do ChangeRole (CatalystProvision, Associate)

```

The policy *membersRoleUpdate* is triggered when a member named *CP* has left the VO and the VO is looking for a new member who can perform the task(s) previously performed by the *CP*. The new member however is initially joined as a temporary member. Later on, the policy *new Catalyst Source Suitable* updates the association of member assigned to fulfil the task *Catalyst Provision* by using *Change Role* action (one of the actions defined in VO-R), which takes as input the task name on which the policy is going to be applied and changes the type of membership from *ExtEntity* to *Associate*.

Criterion C4: Adapt the goal of a VO to meet changing business requirements.

One of the distinguishing characteristics of a VO is its ability to adapt itself with the changing business demands. These changes could range from changing the participants of the VO (which has no effect on its operational model) to adapting the goal of the VO. The life cycle of VBE undergoes *Metamorphism* phase in which the VBE change its purpose completely i.e it changes its domain. Whereas, a VO does not change its goal or domain, it however changes the path to achieve the goal. Hence, a VBE and VO have different set of requirements in this respect.

Evaluation: The VOML Framework also approaches this aspect differently for VBEs and VOs. The modification or changes in the purpose of VBE are reflected by creating new VOs in line with the modified goal(s) of the VBE and terminating older ones that no longer remain relevant with the current goal(s) of the VBE.

At the VO level however, as the goal or purpose of the VOs does not change drastically, only slight modifications are made to the goal that still lie in the realm of current goal, therefore the VOML framework only allows slight modifications to the VOs. Such modifications let only replace or add equivalent competencies such as replacing the means of transportation mechanism from air travel to use of ferries. The VO-R provides *isCapabEquivalent* in its list of conditions that verifies if the two tasks have equivalent capabilities (in which case one task can be replaced with the other). The *ReplaceCapability* and *AddAlternateCapability* actions in its set of actions; the first one replaces one task with other one, whereas, seconds one adds another task as an alternative to the first (but both tasks are still part of a VO).

The *changeSeprationTech* below first checks if the two tasks *ContinuousEquipmentProvider*, *FiltrationEquipmentProvider* are equivalent in terms of capabilities; if yes then the capability of *EquipmentProvision* gets replaced from *FiltrationEquipmentProvider* to *ContinuousEquipmentProvider*.

```

policy changeSeprationTech
appliesTo EquipmentProvision
if isCapabEquivalent(
ContinuousEquipmentProvider,
FiltrationEquipmentProvider )
do ReplaceCapability (EquipmentProvision,
FiltrationEquipmentProvider,
ContinuousEquipmentProvider)

```

Criterion C5: Capture the concept of collaboration by: (a) allowing more than one member to carry out a particular task, and (b) sub-dividing a task into more than one task and assigning each sub-task to be performed by individual members.

A VO is expected to be flexible in its composition and structure so that it can reorganize itself by adding or expelling some members or by dynamically re-assigning tasks to its members; in particular by sharing a task across members or decomposing a complex task into smaller more specialized subtasks each performed by different members.

Evaluation: The VOML framework captures this criterion by offering three types of tasks, (1) *AtomicTask*: which can only be carried out by a single member of VO, (2) *ReplicableTask*: which may be shared by more than one member in order to satisfy the required capacity demands or due to some other demands of the VO, and finally, (3) *ComposableTask*: which can be further sub-

divided into sub tasks each consisting of a subset of capabilities of the original task.

Criterion C6: Preserve the autonomy of members.

The VOs tends to be formed on-demand and the members of the VO need to share resources and skill with each other in order to satisfy the particular goal which the VO promises to achieve. In reality, the members who are partners in one VO might be competitors in some other VO; this puts a strong emphasis on having the members of the VOs freedom to control their own resources and skills and let other members access those resources in a controlled way.

Evaluation: The VOML framework assures this autonomy of members by offering interface style communication where all the required details are provided to the member holding the resource (through tasks) to carry out required action rather then, directly performing actions on the resources provided by some other member. The member holding the resource provides back the outcome of carrying out the task.

Criterion C7: Free from fixed “organizational structure” i.e able to modify the operational model.

Modelling languages for specific domains capture either the business aspect (i.e goal) abstracting away the communication and coordination aspect of the underlying execution environment or they focus the other way round (that the underlying execution environment). It is usually assumed that the business specification model (including coordination and communication model) are not affected by the domain level reorganization of the system (Nguyen, et al. (2006)), (Norman, et al. (2004)), (Patel, *et al.*, (2005)). However any restructuring (reconfiguration) which adds something new, modifies or deletes something from the system does change the coordination and communication model (represented by VO-O). What might be just the matter of adding an “Add” operation to add something new at the structural level, may in fact have the consequence of increase in the number of components or other concrete entities representing the elements of the underlying execution paradigm.

Evaluation: To account for this criterion, the VOML framework keeps the structural model (defined through VO-S sublanguage) and operational model (defined through VO-O sublanguage) separate. Transformation from structural model to operational model is governed by mapping rules. Mapping rules from VO-S to VO-O generate a partial VO-O specification from VO-S specification. The mapping rules also act as conformance verifier of VO-O model to its corresponding VO-S model.

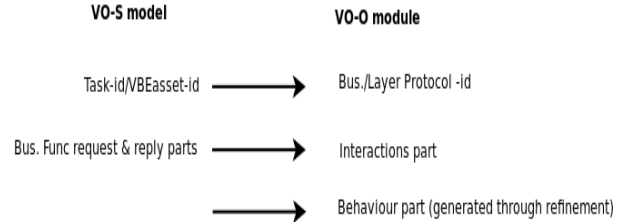


Fig. 1: Relationship between Task/VBEasset and Business/Layer protocols

Fig. 1 offers an overview of how mapping rules guides the transformation of VO-S Tasks and VBEassets into respective VO-O Business and Layer protocols. Empty left-hand side of last arrow reveals the fact that Behaviour part of VO-O protocols manually derived through refinement.

Criterion C8: Define different kinds of relationship between members of a VO and VBE. For example members cooperating or competing over a certain task. Since a VO allows for sharing responsibilities between members if one member is either incapable to do it alone or will take longer to accomplish the task. This naturally leads to have some kind of relationship between those members. Based on the criterion on which the task has been shared between multiple participants the relationship can not only be of cooperation style between them but, the members might in fact be competing with each other.

Evaluation: This requirement has been incorporated by the framework by providing relationship attribute for the ReplicableTasks. The domain of discourse for the relationship attribute can be either cooperation or competition. The (Fig. 2) below specifies a replicable task in which the members are competitors to each other. The criterion in this instance where a VO instance prefer one member over other is that whoever offers less charges for its services will be the part of that VO instance. It is worth mentioning here that each instance of this VO might have different member chosen. The relationship between members performing different subtasks of a ComposableTask is always of type cooperation.

```

ReplicableTask HotelBooking
-----
STRUCTURE
TaskScope
{
  performedBy: VBEParticipant
  allowedMembers : 3
  relationship : competition (lowest, amount)
}
ConfScope
{
  currentState : replicated
  currentMemebrs : 2
}
Competency
{
  capability roomReservation
  {
    resource: rooms
    capacity {totalRooms : 500}
  }
}
    
```

Fig.2: A ReplicableTask description with members competing over lowest cost.

Criterion C10: Resist VOs failing to operate or failing to provide the promised service as long as possible. Though VOs by their very nature are resilient to many failures and resist them as long as possible however, there still are situations when eventually VOs might fail. This criterion refers to the resilience of VOs in hostile environments where a VO might not be able to fulfil its goal how does a modelling language address it. Evaluation: In the following discussion most common causes of VO failure are listed and how does the VOML framework address those. In this regards the limitations of VOML framework with respect to those failures is also discussed, if any. Given below are most frequent causes of the failure:

1. *Contract violation:* Contract violation occurs when the any member of the VO fails to provide the promised service in any way such as by providing resources less in quantity than actually promised, by poor quality of service, by failing to provide on time, etc. The VOML framework does not specifically specify the repercussions of contract violations. However, some policies in VO-R might be a consequence of how some of the contract violations are dealt at the VBE management level. Hence, 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-O levels though and their resolution is at the discretion of VBE and VO management as well.

2. *Unforeseen external event:* Unforeseen and unanticipated changes is a norm in VOs. Failing to tackle those events can lead to the failure of a VO altogether. Even though the VOML framework does not offer provisions to explicitly specify such situations, it is however shown through different scenarios that VOML can handle most of those failures quite naturally. For example if a member of a VO ceases to operate suddenly and the VO has to look for a new member, the tentative members are allowed to join in as temporary members of the VO that have restricted role and/or responsibility until it is decided that the new member is fit for the job. Only then the membership of that member is made permanent in the VO. The only limitation in this case is that the policy describing the event and what to do in such event must be described in advance. So while the VOML framework is rich enough to deal with many different failures, it is the vision and sharpness of management people in the business domain that matters more. The more visionary the management is the more unforeseen situations can be described in advance and the more robust and resilient the VO becomes.

3. *ICT infrastructure failure:* Problems in communication links, such as broken communication

lines or delays in communication lines might fail VOs. Such problems are beyond the realm of VOML framework. However, if appropriate policies are in place then the specific entity (part of VO) that seems to be unreachable by VO due to infrastructure failures/delays and as a result causing the whole VO fail to achieve its goal might be temporarily replaced by other possible temporary members of VO.

4. *Fixed organizational structures:* A VO needs to continuously reconfigure itself to keep aligned with the changing business environment. An operational model satisfying the VO goal with closer to execution environment level details is by virtue limited in its capability to adapt. By far the most easily made reconfiguration on such model is dynamic discovery and binding of components at run-time. The environments for which VOs are appropriate require far more flexibility than that. Hence, a VO can fail if it is not flexible enough to reconfigure itself. The VOML framework provides a solution to 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, and then using mapping guidelines to derive concrete operational description of VOs. This way, a VO defined through the VOML framework provides more flexibility to change different aspects of the VO description at different levels of abstraction leading to more robust and VOs.

4. COMPARISON WITH THE ARCON REFERENCE MODEL

In this section VOML framework is evaluated with respect to the ARCON reference model (Camarinha-Matos, L.M.(2008)). ARCON is a reference model which provides basic elements any manifestation of VOs or VBEs (termed Collaborative Network Organizations (CNOs) in ARCON) possesses. This reference model explores the CNOs in detail and attempts to provide abstract models that capture the most fundamental concepts that CNOs possess.

Hence, evaluating the VOML framework in the light of ARCON reference model would help better understand the VOML framework.

Detailed description of ARCON reference model can be found in (Camarinha-Matos.(2008)). In this section the VOML framework is compared and contrasted with respect to different aspects of the ARCON alongside the brief explanation of ARCON concepts as following:

Life cycle stages: In ARCON, the life cycle stages of CNOs consists of creation, operation, evolution, metamorphism and dissolution. The VOML framework covers, operation and evolution stages of VO life cycle using different modelling languages and by modifying

goal of the VO by changing the capabilities of the tasks. Metamorphism is captured by abandoning current VO and creating new VO. Creation and dissolution stages of life cycle are not currently captured by the VOML framework.

5. CONCLUSION

In this paper the VOML framework has been evaluated against the criteria for evaluating the suitability of any modelling language to model VOs and VBEs with ease and expressiveness. The criteria were discussed in detail in the prequel of the paper.

Furthermore, the VOML framework is evaluated against ARCON reference model as well which provides basic elements that any manifestation of VOs and VBEs exhibit. Comparing it with ARCON further strengthens the VOML framework.

The evaluation identified the strengths and weaknesses of the framework. The paper also discussed how the failures are handled in the framework.

REFERENCES:

- Camarinha-Matos, L. M. and H. Afsarmanesh, (2008) "ARCON reference models for collaborative networks", Editor: Camarinha-Matos, L. M. and H., Afsarmanesh, in Collaborative Networks: Reference Modeling, pages 83-112, Springer US.
- Camarinha-Matos, L. M. and H. Afsarmanesh, (2007) "A framework for virtual organization creation in a breeding environment", Annual Reviews in Control, vol. 31(1) . 119-135
- Camarinha-Matos L. M., H. Afsarmanesh, and M. Ollus. (2005) "Virtual Organizations Systems and Practices", Springer.
- Cummings, J., T. Finholt, I. Foster, and C. Kesselman. (2008) "Beyond being there: A blueprint for advancing the design, development, and evaluation of Virtual Organizations". Technical report, Final Report from Workshops on Building Effective Virtual Organizations In EDUCAUSE, Chicago.
- Foster I. T., (2001) "The Anatomy of the Grid: Enabling Scalable Virtual Organizations". In Proceedings of the 7th International Euro-Par Conference Manchester on Parallel Processing, Euro-Par '01, 1-4, London, UK, Springer-Verlag.
- Hale A. and J. White, (2014), "Developing a framework to establish collaborative enterprise networks" In 8th Annual IEEE Systems Conference (SysCon), 431-438, Ottawa, Canada.
- Nguyen D., S. Thompson, J. Patel, L. Teacy, N. Jennings, M. Luck, V. Dang, S. Chalmers, N. Oren, T. Norman, A. Preece, P. Gray, G. Shercliff, P. Stockreisser, J. Shao, W. A. Gray, and N. Fiddian, (2006), "Delivering services by building and running virtual organizations". BT Technology Journal, vol. 24 141-152, Springer, USA.
- Norman T. J., A. Preece, N. R. Jennings, M. Luck, V. D. Dang, T. D. Nguyen, V. Deora, J. Shao, W. A. Gray, and N. J. Fiddian, (2004) "Agent-based Formation of Virtual Organisations". In Knowledge Based Systems, vol.17, 103-111, Elsevier.
- Patel J., W. T. L. Teacy, N. R. Jennings, M. Luck, S. Chalmers, N. Oren, T. J. Norman, A. Preece, P. M. D. Gray, G. Shercliff, P. J. Stockreisser, J. Shao, W. A. Gray, N. J. Fiddian, and S. Thompson, (2005) "Agent-based virtual organisations for the Grid". In Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, AAMAS '05, 1151-1152, New York, USA, ACM.
- Rajper N. J., (2014) "Criteria for Modelling Languages in the Domain of VOs" Sindh University Research Journal vol. 46 (2) 209-212
- Rajper N.J., (2014) " Modelling Chemical Plant Construction VO in VOML" Sindh University Research Journal vol. 46 (1) 87-92.
- Rajper N. J., (2012) "Virtual Organization Modelling Languages", PhD Thesis, <http://hdl.handle.net/2381/10942>, Leicester, UK.
- Reiff-Marganiec S. and N. J. Rajper (2001) "Modelling Virtual Organizations: Structure and Reconfigurations", In IFIP Advances in Information and Communication Technology, Vol. 362, 297-305, Springer, Boston, US.
- Sánchez N. G., D. Apolinar, G. Zubiaga, J. Atahualpa, I. González and A. Molina (2005) "Virtual Breeding Environment: A First Approach to Understanding Working and Sharing Principles" In Proceedings of InterOp-ESA'05, Springer.
- Sun H., M. Srivatsa , S. Tan, L. Yang, L. M. Kaplan, S. Tao and X. Yan, (2014) "Analyzing Expert Behaviors in Collaborative Networks" In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, 1486-1495, ACM, USA
- Vallejos, R. V., C. P. Lima, and G. Varvakis, (2007) "Towards the development of a framework to create a virtual organisation breeding environment in the mould and die sector", In Journal of Intelligent Manufacturing vol. 18 (5) 587-597 Springer, US