Refinement-Based Techniques in the Analysis of Information Flow Policies for Dynamic Virtual Organisations

Jeremy Bryans, John Fitzgerald, Tom Mccutcheon

Refinement-based Techniques in the Analysis of Information Flow Policies for Dynamic Virtual Organisations

Jeremy W. Bryans[1], John S. Fitzgerald[1] and Tom McCutcheon[2]

[1] Newcastle University, UK
[2] DSTL, UK

Abstract. Dynamic virtual organisations (VOs) can arise in situations in which it is critical that they continue to operate, even in sub-optimal environments. Models of information flow in dynamic VOs are therefore needed in order to permit the rigorous verification of resilience properties before commitments are made to implementation. This paper proposes a refinement-based modelling approach for the design and analysis of VO policy resilience. The approach is demonstrated by using the refinement-based formalism Event-B to model a VO structure, commonly referred to as the Bronze/Silver/Gold structure that frequently arises in multi-agency response to emergencies. Machine-assisted proof is used to compare the validity of alternative information flow policies in Bronze/Silver/Gold when a fault is induced in the VO structure.

Keywords: Information Flow Policies, Refinement-based Modelling, Event-B, Emergency Response, Verification

1. Introduction

Advances in networking technology have made it possible to establish virtual organisations (VOs) of collaborating entities that together enable new capabilities and services that cannot be achieved by the constituent systems alone [1]. In our work, we are particularly concerned with *dynamic* VOs whose membership, structure and goals may change during the life cycle [2] in response to changing circumstances, and particularly faults or malicious attacks. Dynamic VOs bring significant societal and business opportunities by offering, for example, more agile multi-agency emergency response, or coordinated management of resources such as energy.

In order to take advantage of the opportunities presented by dynamic VOs, it is necessary to address the challenges that stem from their complexity and heterogeneity. These include managing the complexity of the behaviours and interactions between participants, managing access to resources, and communicating effectively between diverse stakeholders. Furthermore, as VOs become established and reliance is placed on them, it becomes vital to have methods and tools that support the validation of properties such as safety or security, and the resilience of the VO in maintaining these properties in the face of failures or malicious attack.

To gain confidence in a VO's resilience, it is paramount to have a precise model of its architecture, the resources, functionality and behaviour of the participant systems, infrastructure, and environment [3]. Models support "trade-off" analysis of alternative designs at early development stages, and the precise determination of the contract (rights and responsibilities) that exists between each participant and the VO. Further, the models themselves can assist communication between the diverse stakeholders.

The goal of our research is to provide models and analysis techniques that can be used to support the comparative analysis of alternative VO models, architectures and policies. We use formal modelling notations in order to take advantage of the analytic techniques underpinning these notations, including proof and model-checking. Rather than attempt to prove the properties of a complex VO in a single attempt, there is a strong case for managing the verification as a series of refinement steps starting with a highly abstract description embodying the desired properties of the VO. From this abstract model, detail can be added step by step, with each incremental refinement being verified.

The contribution of this paper is to demonstrate the viability of a refinement-based approach to the modelling and verification of information flow within VOs and to the analysis of alternative VO policy models with respect to resilience. We describe the relevant state of the art in formal modelling of VOs (Section 2) and introduce our proof-of-concept study based on real major incident response procedures in the London area (Section 3). We present a refinement-based model of this VO in Section 4, focussing on response to serious communications failure. Finally we draw conclusions from the modelling approach and discuss further work (Section 5).

2. Formal Modelling of Virtual Organisations

Although there is a perception that formal engineering methods are "very hard to apply" [3], improvements in tools and automation are leading to their increased successful deployment on an industrial scale [4]. Their major benefit is in reducing risk in the development of complex systems such as VOs by allowing early detection and elimination of defects and bottlenecks through the analysis of formal, abstract models using static analysis techniques such as proof and model-checking.

Several characteristics of dynamic VOs – operational and managerial independence of participants, heterogeneity and dynamic change – make the formal modelling of VOs a challenge. A truly comprehensive framework would require abstractions to describe functionality, concurrency, distribution and mobility but few, if any, formal methods currently handle all these features successfully.

Although the formal modelling of VOs is in its infancy, useful models of specific aspects of VOs are being constructed. Recent examples include: the use of the model-oriented formalism Z combined with the CSP process calculus to describe identity management problems [5]; a formal operational model for VO creation [6]; and 'infrastructure-agnostic' formal models of business VOs and Virtual Breeding Environments [7]. The Vienna Development Method (VDM) has been used to model dimensions of VOs with respect to information flow, including structure, membership, responsibility and provenance [8], consistent with (sub-)dimensions

identified by [3]. However, the VDM models are only analysed by means of simulation [9], since proof support for VDM is not yet well supported by tools. The Event-B formalism manages the complexity of proof by structuring models as chains of machines linked by refinement relations, and is well supported by proof tools. Further, Event-B has been demonstrated as a viable basis for modelling conflicts of interest within VOs [10], as well as QoS parameter negotiation [11].

Most work on formal modelling of VOs has concentrated on the description of normal behaviour, and not faults, accidental or malicious. Error detection and recovery can complicate VOs considerably, increasing the risk of design errors, and we expect that formal models could assist the developer of VO policies to assess the impact of faults and the suitability of resilience mechanisms.

The study in this paper aims to investigate the feasibility of using a refinement-based formal method to analyse the resilience of VO structures in advance of deployment. We use Event-B because of its tool support for proof, and its ability to represent the information flow characteristics of a VO. We model a VO that provides opportunities to explore abnormal behaviour because it exists in a volatile environment that may disrupt its operation. We describe the case study in Section 3 and its refinement-based model, including disruption and recovery, in Section 4.

3. Bronze-Silver-Gold: The London Major Incident Procedures

Our case study is based on the London Emergency Service Liaison Panel (LESLP) manual [12], which summarises procedures and responsibilities of the emergency services (e.g., police, fire, or ambulance) in response to a major incident. Activities governed by the manual include rescue and transport of casualties, coordination of resources, and handling release of information to the media. Once a major incident is declared, a VO (which we term a *coalition*) is formed by the services.

The LESLP manual covers many aspects of the coalition life cycle, but we focus on the structural and information flow aspects. These are vital to the successful functioning of the coalition, particularly in managing the release of accurate information, for example regarding the number and location of casualties.

A typical response coalition contains several levels of command: referred to as *bronze, silver,* and *gold* (B/S/G). Each service has members working at each level, so we may refer to "Bronze Police" or "Silver Medic". Silver and Gold commands work in inter-agency coordinating groups. Bronze implements tactics defined by silver. Silver formulates tactics to be adopted by each service following strategy determined by Gold. Gold command, geographically distant, contains the service commanders.

Agents at a common level may communicate freely, but between adjacent levels communication is only between agents of the same service. Information flow out of the coalition is subject to several alternative policies. Certain information may be independently released to the media by an individual service, but casualty figures must be cleared by Gold Police. In our case study, we model an alternative information clearing policy in which all members of the Gold co-ordinating group must clear information for release.

4. Refinement-based Event-B Modelling of Information Flow

An Event-B model [13] is a series of *machines*, each (apart from the first) linked to its predecessor by a refinement relation. A machine contains *variables* modelling state information, *invariants* (labelled *inv*) which constrain the variables, and *events* which change the state. An event has *guards* (labelled *grd*) which must be true for the event to occur, and *actions* (labelled *act*) which change the values of state variables. Each machine may have associated *carrier sets* held in a *context* visible to the machine.

For each step in the refinement chain, a *refinement relation* links the variables in the abstract model with those in the concrete model, and is defined by *gluing invariants*. Adding a machine or a refinement step generates *proof obligations* to ensure the consistency of the machine and the validity of the refinement. For example, the *invariant preservation* obligation ensures that invariants hold after an event occurs. An event in the concrete model refines an event in the abstract model if the concrete guards imply the abstract guards, and the variable states reached after the occurrence of the concrete and abstract events are linked by the gluing invariants.

Proof obligations are generated and in some cases proved automatically by the Rodin tools [14]. Those that are not proved automatically may be discharged by the developer with the help of the theorem proving interface. Proof obligations may be impossible to prove, in which case the model is inconsistent and should be corrected. Models which are proven consistent are said to be *machine-checked*.

Section 4.1 outlines a normative model of the B/S/G coalition[1], including a policy governing the flow of information into, through, and out of the coalition. An additional step refines this policy by distributing responsibility for executing it across several coalition agents. Both policies are machine-checked. The model is then extended with an event modelling the loss of a coalition level, and the policies are re-examined with respect to this enhanced model in Section 4.2.

4.1 Normative Coalition Model and Information Clearing Policies

The model has four levels of abstraction. The most abstract machine treats the coalition as a single entity. The second machine introduces the B/S/G architecture, and the third machine realises this architecture in terms of individual agents. These model a centralised information clearing policy, in which all Gold members simultaneously clear information for release, for example at a joint meeting. Information is learned, cleared for release, and released in separate, atomic events. In the fourth machine, this policy is refined into a set of actions distributed across members of Gold to give a distributed information clearing policy in which information is ready for release when it has been cleared by all members.

Let *INFO* be the set of all information. At the initial level, sets of the *known*, *cleared* for release and *released* information are given as global variables. *Learning, clearing* and *releasing* of information are represented by events. Here, the Event-B model is shown in abbreviated form. *inv2* in Fig. 1 is a key security property of the information flow policy: only cleared information may be released.

[1] The models used in this paper are available at http://deploy-eprints.ecs.soton.ac.uk/

inv1: known, cleared, released $\subseteq$ *INFO*
inv2: released $\subseteq$ *cleared*

LearnNewInfo(i) $\equiv$
grd: i $\in$ *INFO*
act: known := known $\cup$ *{i}*

ClearInfo(i) $\equiv$
grd: i $\in$ *known*
act: cleared :=cleared $\cup$ *{i}*

ReleaseInfo(i) $\equiv$
grd: i $\in$ *cleared*
act: released := released $\cup$ *{i}*

Fig. 1. Learning, clearing and releasing information at the coalition level.

The B/S/G structure is given by the intermediate machine, part of which is shown in Fig. 2. *Bronze*, *Silver* and *Gold* are defined as members of a carrier set *ELEMENT*. The *coalition* variable records the levels currently in the coalition, and *coal_known* records the information known at each level (*inv2*). Separate events describe the creation of a coalition at *Bronze* level and addition of *Silver* and *Gold* in ascending order, as envisaged in [12]. Communication takes place only between adjacent levels. For this study, we consider a policy whereby only *Gold* can clear information for release (*ClearInfoGold*). The information must be known by *Gold* (*grd1*) and not previously cleared (*grd2*). In *ReleaseInfoGold, grd3* ensures that information may be released only after it is cleared.

inv1: coalition $\subseteq$ *{Bronze, Silver, Gold}* *inv2: coal_known* $\in$ *coalition* $\leftrightarrow$ *INFO*

ClearInfoGold(i) $\equiv$
grd1: i$\in$ *coal_known[{Gold}]*
grd2: i$\notin$ *cleared*
act1: cleared :=cleared $\cup$ *{i}*

ReleaseInfoGold(i) $\equiv$
grd1: i$\in$ *coal_known[{Gold}]*
grd2: Gold $\in$ *coalition*
grd3: i $\in$ *cleared*
act1: released := released $\cup$ *{i}*

Fig. 2. The architectural-level machine.

The next level introduces agents. An agent's service (*Police*, *Fire* or *Ambulance*) and level are given by the functions *service* and *element* respectively. The relation *info* gives the information known by individual agents (*inv1* in Fig. 3). *inv2* and *inv3* are the gluing invariants between *info* and *coal_known*. They ensure that information known by an agent in *info* must be known at the level of the agent in *coal_known* and vice versa. *ClearInfoGoldAllElements* is the atomic clearing event in this machine.

inv1: info $\in$ *AGENT* $\leftrightarrow$ *INFO*
inv2: $\forall$ *i,ag* $\bullet$ *ag* $\in$ *dom(service)* $\wedge$ *i*$\in$ *info[{ag}]* $\Rightarrow$ *element(ag)* $\mapsto$ *i* $\in$ *coal_known*
inv3: $\forall$ *lvl,i* $\bullet$ *lvl*$\in$ *dom(coal_known)* $\wedge$ *lvl* $\mapsto$ *i* $\in$ *coal_known* $\Rightarrow$
($\exists$ *ag* $\bullet$ *ag*$\in$ *dom(element)* $\wedge$ *element(ag)=lvl* $\wedge$ *ag* $\mapsto$ *i* $\in$ *info)*

ClearInfoGoldAllElements(i) $\equiv$
grd1: element~[{Gold}] $\neq$ *{}*
grd2: $\forall$ *a* $\bullet$ *(a* $\in$ *dom(element)* $\wedge$ *element(a)=Gold)* $\Rightarrow$ *i* $\in$ *info[{a}]*
grd3: i$\notin$ *cleared*
act1: cleared = cleared $\cup$ *{i}*

Fig. 3. The agent-level machine.

The fourth machine refines the centralised clearing policy into a distributed one (Fig. 4). Any services represented at *Gold* individually clear information for release (event *ClearInfoService*), and when all services represented at *Gold* have cleared information, it can be cleared by the coalition (event *ClearInfoGoldDistrib*). We also include the consistency invariant (*inv1*) between the information cleared by the coalition as a whole (variable *cleared*) and the information cleared by each service (variable *serviceCleared*). This records the intuition that all information cleared by the coalition must have been cleared by all members of *Gold*. This requires a change to the event which adds an agent to the coalition: if an agent joins *Gold* it must agree with the information clearing decisions already made by the coalition.

$inv1: \forall ag \bullet ag \in dom(service) \Rightarrow$
$\quad (element(ag)=Gold \Rightarrow cleared \subseteq serviceCleared[\{service(ag)\}])$

$ClearInfoService(ag,i) \equiv$
$grd1: ag \in dom(element)$
$grd2: element(ag) \in Gold$
$grd3: i \notin serviceCleared[\{service(ag)\}]$
$act1: serviceCleared := serviceCleared \cup \{service(ag) \mapsto i\}$

$ClearInfoGoldDistrib(i) \equiv$
$grd1: element\sim[\{Gold\}] \neq \{\}$
$grd2: \forall a \bullet (a \in dom(element) \wedge element(a)=Gold) \Rightarrow i \in info[\{a\}]$
$grd3: \forall a \bullet (a \in dom(element) \wedge element(a)=Gold) \Rightarrow i \in serviceCleared[\{service(a)\}]$
$grd4: i \notin cleared$
$act1: cleared = cleared \cup \{i\}$

Fig. 4. The distributed clearing policy.

4.2 Analysis of Policy Resilience

A significant risk to the coalition is the loss of a command layer, for example through communications failure. We consider the case where *Gold* is lost and must be replaced by a new set of agents, and the effect this has on the centralised and distributed information clearing policies. Since it is possible that the level lost was the sole holder of certain information, possible information loss must be included at each abstraction level in the model. In the first machine, this is modelled by the event *LoseInfo* (Fig. 5). We do not give the event in the intermediate machine. The event which models the loss of an entire level of command in the third machine is *RemoveLevelTotal*.

$LoseInfo(inf) \equiv$
$grd: inf \subseteq INFO$
$act: known := known \setminus inf$

$RemoveLevelTotal(lvl) \equiv$
$grd1: lvl \in ran(element)$
$act1: coalition := coalition \setminus \{lvl\}$
$act2: service := element\sim[\{lvl\}] \lhd service$
$act3: element := element\sim[\{lvl\}] \lhd element$
$act4: info := element\sim[\{lvl\}] \lhd info$

Fig. 5. Removing a level of command.

All obligations resulting from the addition of these events can be proved, although three extra invariants are required in the third machine. However, the distributed clearing policy causes a problem: it is now impossible to prove *inv1* in Fig. 4. The offending event is *CreateGold*. If Gold has been lost and is being recreated, we cannot show that the agent re-forming Gold belongs to a service that is prepared to clear all information that has been cleared by the previous instantiation of Gold.

Investigation of the failed proof, offending event and consistency invariant suggests several ways forward. The policy could be extended to require that an agent re-forming Gold agrees with all previous clearing decisions. This would entail altering the event *CreateGold*. Alternatively, the consistency requirement between the two policies could require that any released information was once cleared by all services that made up *Gold*. This would entail maintaining a record of the membership of *Gold* at all times, and the information cleared by each *Gold* grouping. A further alternative is to treat the creation of Gold as different from its recreation, and the set of information cleared by the coalition could be reset to empty. The choice between these alternatives would rest with the policy developer and stakeholders. The advantage of the formal model is that the precise cause of the inconsistency is quickly identified and can be communicated to stakeholders, along with the avenues for re-design.

5. Discussion

This paper has shown how a formal model of a virtual organisation can provide a good basis for an analysis of the information flow policy of the organisation. When potential defects came to light, several options for alternative designs were readily identified.

Many topics are open for further research. Extensions to the model are possible, such as considering the response to multiple intersecting crises. The information model could be refined to include, for example, policies for handling classified information. Modelling the roles and the associated responsibilities of the agents would also lead to richer possibilities for policy analysis, including policies other than information flow. Many emergency response coalitions are much more ad-hoc than B/S/G, and looking at ways to model these less structured ways of dealing with complex crises is also of interest, including ways to integrate formal and semi-formal approaches [3,15].

Acknowledgements

The authors' work is supported by the EU FP7 Integrated Project DEPLOY (www.deploy-project.eu) and the EPSRC Platform Grant TrAmS.

References

1. Camarinha-Matos, L.M., Afsarmanesh, H., Garita, C., Lima, C.: Towards an architecture for virtual enterprises. Journal of Intelligent Manufacturing, vol. 9, pp. 189-199. Chapman & Hall (1998)
2. Dimitrakos, T., Golby, D., Kearney, P.: Towards a Trust and Contract Management Framework for Dynamic Virtual Organisations. In: eAdoption and the Knowledge Economy: eChallenges 2004, pp. 27-29. Kluwer Academic (2004)
3. Camarinha-Matos, L.M., Afsarmanesh, H.: A comprehensive modeling framework for collaborative networked organizations. Journal of Intelligent Manufacturing, vol. 18, pp. 529-542. Springer (2007)
4. Woodcock, J.C.P., Larsen, P.G., Bicarregui, J., Fitzgerald, J.S.: Formal methods: Practice and experience, Computing Surveys 41(4), pp. 1-36. ACM (2009)
5. Haidar, A.N., Coveney, P.V., Abdallah, A.E. et al.: Formal Modelling of a Usable Identity Management Solution for Virtual Organisations. In: Bryans, J.W., Fitzgerald, J.S. (eds.): Proc. 2nd Workshop on Formal Aspects of Virtual Organisations, Eindhoven, November 2009, Electronic Proceedings in Theoretical Computer Science, vol. 16, pp. 41-50 (2010)
6. McGinnis, J., Stathis, K., Toni, F.: A Formal Framework of Virtual Organisations as Agent Societies. In: Bryans, J.W., Fitzgerald, J.S. (eds.): Proc. 2nd Workshop on Formal Aspects of Virtual Organisations, Eindhoven, November 2009, Electronic Proceedings in Theoretical Computer Science, vol. 16, pp. 1-14 (2010)
7. Bocchi, L., Fiadeiro, J., Rajper, N., Reiff-Marganeic, S.: Structure and Behaviour of Virtual Organisation Breeding Environments. In: Bryans, J.W., Fitzgerald, J.S. (eds.): Proc. 2nd Workshop on Formal Aspects of Virtual Organisations, Eindhoven, November 2009, Electronic Proceedings in Theoretical Computer Science, vol. 16, pp. 26-40 (2010)
8. Bryans, J.W., Fitzgerald, J.S., Jones, C.B., Mozolevsky, I.: Formal Modelling of Dynamic Coalitions, with an Application in Chemical Engineering, Proc. IEEE Intl. Symp. on Leveraging Applications for Formal Methods, Cyprus, 2006, pp. 91-98. IEEE (2007)
9. Fitzgerald, J.S., Bryans, J.W., Greathead, D., Jones, C.B.: Animation-based Validation of a Formal Model of Dynamic Virtual Organisations. In Boca, P., Bowen J.P., Larsen, P.G.(eds.): Proc. BCS-FACS Workshop on Formal Methods in Industry, Electronic Workshops in Computing. British Computer Society (2008)
10. Arenas, A., Aziz, B., Bicarregui J.C., Matthews, B.: Managing Conflicts of Interest in Virtual Organisations, Electronic Notes in Theoretical Computer Science, 197(2), Proc. 3rd Intl. Workshop on Security and Trust Management (STM 2007), pp. 45-56. (2008)
11. Belhay, H., Balouki, Y., Boudadi, M., amd ElHajji, S.: Using Event B to specify QoS in ODP Enterprise Language. In Camarinha-Matos, L., Boucher, X. and Afsarmanesh, H. (eds): Proc. 11th IFIP Working Conference on Virtual Enterprises (PRO-VE 2010), pp. 478-485, IFIP 2010
12. London Emergency Services Liaison Panel: Major Incident Procedure Manual. Seventh Edition (2007). Available at www.met.police.uk/leslp.
13. Abrial, J-R. Modelling in Event-B: System and Software Engineering. Cambridge University Press. (2010)
14. Abrial, J-R, Butler, M., Hallerstede, S., and Voisin, L.: An Open Extensible Tool Environment for Event-B. In Liu, Z. and He, J. (eds.): Formal Methods and Software Engineering, LNCS, vol. 4260, pp. 588-605. (2006)
15. Fitzgerald, J.S., Bryans, J.W.: The Verifiable Virtual Organisation: a Position Paper. In Proc. Formal Aspects of Virtual Organisations 2008, Technical Report CS-TR-1098, School of Computing Science, Newcastle University. (2008)