Formal Modelling of Dynamic Coalitions, with an Application in Chemical Engineering

Jeremy W. Bryans*, John S. Fitzgerald*, Cliff B. Jones*, Igor Mozolevsky*
*School of Computing Science, Newcastle University, UK
Email: firstname.lastname@newcastle.ac.uk

***Abstract*— Dynamic coalitions are temporary alliances formed between agents in order to achieve specific business goals. Such coalitions can vary widely in architecture, scale, complexity and lifetime. Few techniques have so far emerged to assist in the analysis and design of coalitions. We apply formal model-oriented techniques to help structure the space of dynamic coalitions, with an emphasis on modelling information flow. A series of models is developed in VDM, each emphasising a different "dimension" of the space. These are used to characterise a new dynamic coalition architecture under development for the chemical engineering industry. Tool-supported analysis of this formal model has identified potential improvements in the coalition architecture.**

I. Introduction

Improvements in the capabilities of networking and ambient computing technologies enable individuals and organisations to form *dynamic coalitions*. These are temporary alliances, driven by a desire to cooperate towards a common goal. For example, in dynamic business environments [1], companies may form a coalition to capitalise on a market opportunity. In disaster response scenarios, emergency services, military units and civil organisations must work together to mitigate the impact of a dangerous incident [2].

The ability to analyse information flow, security, privacy and trust in dynamic coalitions is particularly significant in many applications. However, the architects of such coalitions currently lack a basis on which to evaluate at design-time the likely consequences of the decisions that they make regarding coalition architecture and policies. The long-term aim of our work is to leverage formal modelling technology to help provide such a basis. In this paper, we show how formal modelling of particular dimensions of dynamic coalitions can help towards this goal, illustrating this with an application in the chemical engineering industry.

It is worth briefly identifying the two practitioner groups who are our "customers". Colleagues in the UK Defence Science and Technology Laboratory wished to develop formal tools to assist in considering information flow in defence-related coalitions. Other colleagues in the chemical engineering industry (via the GOLD project[1]) are developing middleware to support coalitions that form around the development of novel chemical compounds. Both groups need tools for analysing security and access control in different dynamic coalition structures, and both encouraged a formal approach in order to form a sound basis for the development of analytic tools.

Our approach is constructive. Since there is such a wide range of possible coalition architectures, we first map out a space of possible coalition structures using formal modelling techniques. We then validate the approach with our customers, by applying it to a real dynamic coalition structure in the chemical engineering industry, using it to help analyse access control policies. Our future work will address the development of stronger tools supporting analysis of our models.

We map the space of dynamic coalitions by identifying several "dimensions", each corresponding to some aspect of the coalition that may be significant to its architect, e.g. membership policy or information transfer. In each dimension, we distinguish *information* (the material traded between agents in a coalition) from *meta-information* (information *about* the agents, coalitions or information itself, such as the age of a piece of information, or the identity of a coalition). Generally, each dimension that we explore corresponds to a form of meta-information and the models that we develop make that meta-information explicit. In each model, consideration of the invariants, preconditions etc. leads to alternative models representing design choices. The result of this analysis is a suite of models that deal with individual dimensions and present the coalition architect with a range of design alternatives, allowing a particular architecture to be placed within the space of coalitions.

There are several taxonomies of dynamic coalitions, mainly in management science. Some categorise coalitions by structure of information flow [3], and some use dimensions such as strategy, process, structure, knowledge and culture [4]. Much attention is paid to the network security and access control in military dynamic coalitions [5], [6], [7]. However, existing approaches do not provide a formal basis for analysing coalition properties during design and operation.

Our motivation is to support the architect in designing coalition structures. We have therefore selected dimensions that cover a large space of possible dynamic coalitions, and which present the architect with design alternatives. We do not aim to provide a comprehensive taxonomy, but we have validated the approach and the dimensions selected, by showing their applicability to the GOLD architecture, using the models to help identify areas of incompleteness and inconsistency in existing designs. We would expect the range of dimensions selected, and their associated models, to be extended in future.

[1] `http://www.goldproject.ac.uk/`

We have used formal modelling technology based on the Vienna Development Method (VDM) [8], [9], [10]. VDM, like Z [11] and B [12], uses a model-oriented language that emphasises the modelling of data, state and functionality, making it suitable for describing the functionality involved in forming and operating dynamic coalitions. The analysis of other aspects, such as the purposes and goals of coalitions, would of course require the use of complementary formalisms. VDM has been used extensively to model computer-based systems, especially with domain experts [13], and benefits from strong tool support.

In this paper we introduce the main dimensions of dynamic coalitions that have been identified, the formal models that describe them and their application in chemical engineering. We begin with a basic model for coalitions that have a membership dimension (Section II). We then explore the models of information storage (Section III), and communication (Section IV). Section V briefly examines the dimensions of authorisation structure, provenance, time and trust. In Section VI we describe our application in the chemical engineering industry, and the associated tool support. Section VII draws conclusions from the work conducted so far and identifies further research. For reasons of brevity we show only selected aspects of the modelling work here. The dimensions and associated models are described in [14][2]. The formal notation used is ISO Standard VDM-SL [15] mathematical syntax, with some "sugaring" to ease presentation.

II. Coalitions and Coalition Membership

Our basic model is of a *global state* composed of *Agents* which may join and leave groupings known as *Coalitions*. When we consider each aspect of a coalition (membership, information transfer, provenance, trust etc.) we have to consider where relevant data lies in the system: at the global, coalition or agent level. This tripartite structure is present throughout the work reported in [14].

The types Aid and Cid represent possible agent and coalition identifiers respectively. Elements of both types are structureless *tokens*. In this paper introduced types will be tokens unless explicitly defined otherwise. We will introduce a type $Agent$ to represent agents. Our basic state is formally defined as follows:

$$\begin{array}{lll}\Sigma :: & coals & : Cid \xrightarrow{m} Aid\text{-}\mathbf{set} \\ & agents & : Aid \xrightarrow{m} Agent\end{array}$$

$$\mathbf{inv}\ (coals, agents) \triangleq (\bigcup \mathbf{rng}\ coals) \subseteq \mathbf{dom}\ agents$$

The state Σ consists of two components: $coals$ records an association (in VDM, a mapping) between coalition identifiers and the sets of (identifiers of) agents that are members of the coalition; the $agents$ component relates agent identifiers to agents. The invariant is a predicate ensuring that the agent identifiers in coalitions are all genuine, i.e. they are known in the $agents$ component. Note that, in this model, we allow the same set of agents to be participating in two different coalitions, allowing for different structures, membership schemes etc.

Membership meta-information is a relation between agents and coalitions. It is already present in the Σ model, so it is used as a starting point. A key question is: where does responsibility lie for performing the operations that join and remove agents to and from coalitions? The model Σ takes a coalition-oriented view of membership in the sense that the agent identifiers are associated with their individual coalitions. In order to describe the act of joining an agent to a coalition, we give an operation specification:

$$\begin{array}{l}Join\ (a{:}\ Aid, c{:}\ Cid) \\ \mathbf{ext\ wr}\ coals\ :\ Cid \xrightarrow{m} Aid\text{-}\mathbf{set} \\ \quad \mathbf{rd}\ agents\ :\ Aid \xrightarrow{m} Agent \\ \mathbf{pre}\ a \in \mathbf{dom}\ agents \wedge c \in \mathbf{dom}\ coals \wedge a \notin coals(c) \\ \mathbf{post}\ coals = \overleftarrow{coals} \dagger \{c \mapsto \overleftarrow{coals}(c) \cup \{a\}\}\end{array}$$

The precondition in the *Join* operation records the assumption that the agent and coalition are both known, and that the agent is not already a member of the coalition. The *Remove* operation performs the inverse:

$$\begin{array}{l}Remove\ (a{:}\ Aid, c{:}\ Cid) \\ \mathbf{ext\ wr}\ coals\ :\ Cid \xrightarrow{m} Aid\text{-}\mathbf{set} \\ \mathbf{pre}\ c \in \mathbf{dom}\ coals \wedge a \in coals(c) \\ \mathbf{post}\ coals = \overleftarrow{coals} \dagger \{c \mapsto \overleftarrow{coals}(c) \setminus \{a\}\}\end{array}$$

Both of these operations require that the coalition already exists prior to the addition or removal of a member.

If authorisation to join a coalition is an important factor, this dimension may be elaborated. For example, support for joining and leaving decisions may have to be gathered from more than a certain threshold of existing coalition members. This threshold value must be recorded within the coalition structure. If we assume it to be value between zero and one, we get model Σ_{auth}:

$$\begin{array}{lll}\Sigma_{auth} :: & coals & : Cid \xrightarrow{m} Coalition \\ & agents & : Aid \xrightarrow{m} Agent\end{array}$$

$$\mathbf{inv}\ (coals, agents) \triangleq \bigcup\{c.members \mid c \in \mathbf{rng}\ coals\} \subseteq \mathbf{dom}\ agents$$

$$\begin{array}{lll}Coalition :: & members & : Aid\text{-}\mathbf{set} \\ & threshold & : \mathbb{R}\end{array}$$

$$\mathbf{inv}\ (\text{-}, threshold) \triangleq 0 \leq threshold \wedge threshold \leq 1$$

Yet more elaborate membership authorisation schemes may be envisaged and modelled. For example, the model could itself be generic with a set of parameters governing membership determination.

III. Information

In order to model the flow of information in dynamic coalitions, it is necessary to decide on a representation for

[2]The models developed are available at `http://www.dirc.org.uk/resources/dc.html`

information, its storage and creation within the model. This section explores these dimensions, while Section IV deals with information flow itself.

Several significant abstraction decisions have to be made regarding information. We will use the term "information" in a general sense to describe the data traded between agents in a coalition, and between coalitions and their environment. Given that the purpose of the model to analyse information flow, rather than accuracy of information with respect to some external world, we will also refrain from attempting to model this semantic relationship.

Representing and Identifying Information: Models of specific coalitions may choose common information representation frameworks. However, for the purposes of modelling information flow, the models we develop here are neutral about the particular representation chosen. We will use the data type $Information$ to stand for the chosen representation, and treat this as a collection of unstructured tokens.

Information is an unusual kind of resource in that it may be copied and modified arbitrarily, as well as transferred. If an information item is copied, and we wish to distinguish the copy from the original, it is necessary to identify each item by means of a key. In this case, where individual items have unique keys as identifiers, it is necessary to maintain a mapping from the keys to the information values (formally, $InfoKey \xrightarrow{m} Information$). An alternative is to regard $Information$ items as unkeyed values and so to regard the collection of information as an $Information$-**set**. There are advantages for the mapping model because one can discuss, for example, the visibility of information in terms of the sets of $InfoKey$. However, this approach makes the discussion of copying more difficult. Another reason for preferring the set structure is that it is always possible to embed a "key" within $Information$, although this has to be done with an awareness of "normal forms" and any requirements for uniqueness. The decision about whether to use a mapping-based or set-based model is likely to vary between applications. We illustrate both approaches. In the first part of the paper we represent information as sets of tokens, and in the case study in Section VI we use a mapping to represent the information.

Locating Information in the Model: Taking the basic model Σ as a starting point again, we can envisage information being held at agent, coalition and global levels, as was the case for membership. At the agent level, the $Agent$ data type can be augmented with an information store. We choose to represent this using a set of $Information$ tokens. The resulting definition of $Agent$ would be:

$$Agent :: info : Information\text{-}\mathbf{set}$$

Shared information, common to members of a coalition, would reside at the coalition level. The coalition model is therefore more than just the set of member identifiers, and has its own information set. We introduce the type $Coalition$ to model this:

$$\begin{aligned} Coalition :: \; & info : Information\text{-}\mathbf{set} \\ & \cdots : \cdots \end{aligned}$$

Global common knowledge, shared across the space of all coalitions, is at the outermost level, leading to a new model:

$$\begin{aligned} \Sigma_{loc} :: \; & info : Information\text{-}\mathbf{set} \\ & coals : Cid \xrightarrow{m} Coalition \\ & agents : Aid \xrightarrow{m} Agent \end{aligned}$$

We have not added an invariant that information stored at the global or coalition levels should be present within agents. This allows us to model problematic situations in which a global view is not shared by all the participating agents. We have opted for what is intended to be an intuitive model here. Other models, such as only allowing information to be stored within agents, are possible.

Creating and Sharing Information: Here we consider the basic operations of information creation and sharing "up the hierarchy" that can be described over the Σ_{loc} model. If we choose not to model the source of information explicitly, we must include within agents an operation such as *Discover*, which adds new set of $Information$ tokens to the agent's knowledge base:

$$\begin{aligned} & Discover \; (a{:}\, Aid, is{:}\, Information\text{-}\mathbf{set}) \\ & \mathbf{ext\ wr} \; agents : Aid \xrightarrow{m} Agent \\ & \mathbf{pre} \; a \in \mathbf{dom}\, agents \\ & \mathbf{post} \; agents = \overleftarrow{agents} \dagger \\ & \qquad \{a \mapsto \mu(\overleftarrow{agents}(a), info \mapsto \overleftarrow{agents}(a).info \cup is)\} \end{aligned}$$

Since the model is intended for the analysis of information transfer and is not concerned with modelling the external environment, this operation is almost trivial, simply allowing information to appear in the model. The operation is defined at the agent level (it works on the information store of a specific agent). One might envisage a similar operation being available at the coalition level, if the coalition has its own ability to acquire information independent of the participating agents. It is somewhat harder to motivate a version of the operation at the global level.

The layered model permits the definition of operations describing the movement of information from the agent level to coalition level, a form of sharing:

$$\begin{aligned} & Share \; (a{:}\, Aid, c{:}\, Cid, is{:}\, Information\text{-}\mathbf{set}) \\ & \mathbf{ext\ wr} \; coals : Cid \xrightarrow{m} Coalition \\ & \quad \mathbf{rd} \; agents : Aid \xrightarrow{m} Agent \\ & \mathbf{pre} \; a \in \mathbf{dom}\, agents \wedge is \subseteq agents(a).info \wedge \\ & \qquad c \in \mathbf{dom}\, coals \wedge a \in coals(c).members \\ & \mathbf{post} \; coals = \overleftarrow{coals} \dagger \\ & \qquad \{c \mapsto \mu(\overleftarrow{coals}(c), info \mapsto \overleftarrow{coals}(c).info \cup is)\} \end{aligned}$$

A similar operation could promote information from coalition to global level, placing it in the $info$ field of Σ_{loc}.

Losing Information: When an agent leaves a coalition, the information held within the agent may be lost to the coalition. A protocol might be employed which requires the sharing of certain information (placing at the coalition level) before

permission for departure is granted. In the opposite direction, the agent may be able to copy coalition-level information into its individual store prior to departure from the coalition. At coalition dissolution, information stored at the coalition level could be deleted or migrated up to the global level, or copied or distributed among the agents in the former coalition.

If the information stores are shared repositories, it is possible to model agents losing access to information but this raises questions about which part of the store the departing agent had accessed and when. These questions would be particularly pertinent if the departing agent may have hostile intent. Countermeasures here include changing information so that the knowledge itself is altered so that the departing agent's knowledge is no longer meaningful. This is practiced regularly with such information such as group keys.

IV. Information Transfer

In this section, we consider the options for movement of information between agents. The models developed in this section describe the functionality of information transfer on the basis of the meta-information about membership and information discussed in previous sections. At this level, the model is not concerned with mechanisms of transmission, so much as the preconditions: who can participate in an information transfer, and what can be transmitted? As a starting point, we extend the original base model Σ to include the information stored in each agent as a set of $Information$ tokens.

$$\begin{array}{l} \Sigma_{simp} :: \quad coals : Cid \xrightarrow{m} Aid\text{-}\mathbf{set} \\ \qquad\qquad agents : Aid \xrightarrow{m} Agent \\ \mathbf{inv}\ (coals, agents) \triangleq (\bigcup \mathbf{rng}\ coals) \subseteq \mathbf{dom}\ agents \end{array}$$

$$Agent = Information\text{-}\mathbf{set}$$

It is likely that individual agents will operate policies regarding the clearing of information for transfer. In military contexts this is implemented via a system of classification levels for information and clearance levels for personnel: documents can only be read if the clearance level of the reader is at least as high as the classification of the document. At a high level of abstraction, this could be considered as a mapping from each piece of $Information$ to the set of potential recipients. This is modelled within the extended $Agent$ type below as the mapping "*clearance*":

$$\begin{array}{l} Agent :: \quad info : Information\text{-}\mathbf{set} \\ \qquad\qquad clearance : Information \xrightarrow{m} Aid\text{-}\mathbf{set} \\ \mathbf{inv}\ (info, clearance) \triangleq \forall i \in \mathbf{dom}\ clearance \cdot i \in info \end{array}$$

$$\begin{array}{l} \Sigma_{a\text{-}tr} :: \quad coals : Cid \xrightarrow{m} Aid\text{-}\mathbf{set} \\ \qquad\qquad agents : Aid \xrightarrow{m} Agent \\ \mathbf{inv}\ (coals, agents) \triangleq \bigcup \mathbf{rng}\ coals \subseteq \mathbf{dom}\ agents \wedge \\ \quad \bigcup\{\bigcup \mathbf{rng}\ ags.clearance \mid ags \in \mathbf{rng}\ agents\} \subseteq \\ \quad\quad \mathbf{dom}\ agents \end{array}$$

The extra invariant clause on the type $Agent$ asserts that all the information which may be revealed by an agent is known by that agent. The second conjunct of the invariant on $\Sigma_{a\text{-}tr}$ asserts that all the agents to whom information may be revealed are valid.

The clearance component can be used to govern information transfer:

$$\begin{array}{l} InfoTransfer\ (from, to: Aid, is: Information\text{-}\mathbf{set}) \\ \mathbf{ext\ wr}\ agents : Aid \xrightarrow{m} Agent \\ \mathbf{pre}\ \{from, to\} \subseteq \mathbf{dom}\ agents \wedge \\ \quad \forall i \in is \cdot to \in agents(from).clearance(i) \wedge \\ \quad is \subseteq agents(from) \\ \mathbf{post}\ agents = \overleftarrow{agents} \dagger \\ \quad \{to \mapsto \mu(\overleftarrow{agents}(to), info \mapsto \overleftarrow{agents}(to).info \cup is)\} \end{array}$$

Alternative policies may be pursued, for example returning an explicit error if the clearance precondition is not satisfied, or transferring the cleared subset of is: $is \setminus \{i \in is \mid to \notin agents(from).clearance(i)\}$. In this generic model, we do not exclude certain forms of communication such as self-to-self or sharing information that may already be present at coalition or global levels, since we may wish to analyse the consequences of these behaviours.

V. Other Dimensions

In this section we present a selection of other dimensions in outline only. In [14] we discuss these and further dimensions in more detail.

Governance: Various "structures" have been proposed for dynamic coalitions, mostly limited to information transfer (e.g. [3]). For example, a *star* structure has a single agent as the nexus for all communication in the coalition; a *tree* structure has local star structures passing information up from centres to the next level. Our modelling work suggests that information transfer structures describe just part of a coalition's character. A key aspect of a coalition's structure is its governance: which agents may authorise specific acts such as information transfer or membership operations. Our use of pre/postcondition specifications for operations governing membership and information transfer emphasises that these operations may require permission. Often, authorisation structures in a dynamic coalition will exist only as a consequence of rights, obligations and privileges which are agreed, known and held within the coalition. The explicit, early consideration of these structures is likely to have a significant effect on the design of a robust coalition.

In this dimension, we consider authorisation structures explicitly and independent of the operations being authorised. The meta-information expressed in the models of governance concerns the authorisation structures which may be added to any of the models developed so far, allowing forms such as star and tree structures to be represented for authorisation as well as communication.

Operations, such as those performing information transfer, which share a common authorisation structure, may require authorisation from one or more specific agents. We thus consider an authorisation structure as a relation between agents.

Formally, we could define a data type representing such a relation:

$$AuthRel = (Aid \times Aid)\text{-}\mathbf{set}$$

The presence of a pair (a_1, a_2) in a relation indicates that a_1 is capable of authorising an operation by a_2. Properties of this data-type can be specified with invariants. Given such a general model, it is possible to describe common structures by means of combinations of conditions on the relation. For example, a tree-based authorisation structure is characterised by a constraint that every agent that is subject to authorisation is subject to only one authoriser.

Modelling Provenance: In a dynamic coalition, it will often be important for an agent to know the source of the information it holds (its provenance), as well as the information itself. A simple model is for each agent to associate with each item of information the agent from whom it was received. We can alter the information record within an agent to associate a single providing agent with each information token. In a more realistic model, when information is transferred, the provenance information is passed with it. Agents would build up a list of agents, representing the path that the information has taken to them, as in the model Σ_{p^*}.

$$\begin{array}{lrcl} \Sigma_{p^*} :: & coals & : & Cid \xrightarrow{m} Aid\text{-}\mathbf{set} \\ & agents & : & Aid \xrightarrow{m} Agent \end{array}$$

$$Agent :: agentinfo : TrackedInformation\text{-}\mathbf{set}$$

$$\begin{array}{lrcl} TrackedInformation :: & item & : & Information \\ & prov & : & Aid^* \end{array}$$

Such a model immediately raises the possibility of an agent lying about the provenance information it passes on, and the (in)consistency of corresponding provenance trails. We could include an invariant to stipulate that provenance information is passed on accurately and in full, if this was appropriate to the coalition.

Trust: The problem of how to infer trust from meta-information is still open[3], and proposed solutions are necessarily context dependent. After trust has been computed the question of how to use it has many answers, again context dependent. We do not consider these questions, and begin by assuming that trust values have been obtained (represented as an ordered type $Trustvalue$). These trust values may represent an agent's trust in other agents, or an agent's trust in information.

An agent's trust in other agents can be represented as

$$Agent :: aTrust : Aid \xrightarrow{m} Trustvalue$$

Over time, the *aTrust* mapping will be updated, according to some set of rules that an agent has. For example, if an agent a applies a "Friend-of-a-friend" rule, and agent b (whom a trusts) itself trusts agent c, then agent a may be inclined to trust agent c as well.

[3]This is evidenced by the TrustCom project and the related iTrust conferences: `http://www.eu-trustcom.com`

Coalition members might be expected to trust the other members of the coalition to some degree. If this were the case, it could be mandated by a suitable invariant:

$$\begin{array}{l} \Sigma_{trust\text{-}c} :: \quad coals : Cid \xrightarrow{m} Coalition \\ \qquad\qquad\quad agents : Aid \xrightarrow{m} Agent \\ \textbf{inv}\ (coals, \text{-}) \triangleq \\ \quad \forall c \in \textbf{dom}\ coals \cdot \forall m, m' \in coals(c).members \cdot \\ \qquad m.aTrust(m') \geq coals(c).cTrust \end{array}$$

$$\begin{array}{lrcl} Coalition :: & members & : & Aid\text{-}\mathbf{set} \\ & cTrust & : & Trustvalue \end{array}$$

Expiry Times: Some information "expires": it goes out of date at a certain time. In the simple agent model below, a single time is associated with each piece of information, with a special value "**nil**" to denote that a piece of information is always valid. At the recorded time, the value of the information changes in some quantifiable way: for example, a document may move from "classified" to "unclassified", or meteorological data may change from "current" to "out-of-date".

$$\begin{array}{lrcl} Agent :: & agentinfo & : & Information\text{-}\mathbf{set} \\ & current\text{-}time & : & Time \end{array}$$

$$\begin{array}{lrcl} Information :: & item & : & token \\ & expire & : & [Time] \end{array}$$

Using *expire*, an agent can check if a piece of information is still valid at a particular time.

$$still\text{-}valid : Information \times Time \rightarrow \mathbb{B}$$

$$still\text{-}valid(info, time) \triangleq info.expire = \textbf{nil} \vee info.expire \geq time$$

It is possible that collated information may remain valuable for longer than any of its elements. This can be modelled in a straightforward way. Further, we describe here predictable expiry times. If information could also expire at the occurrence of an unpredictable event we would need to refine the model.

VI. Case Study: The GOLD Architecture

This section outlines a case study conducted in order to help validate the modelling work described above (see [14] for further detail). We describe the virtual organisation architecture under development within the GOLD project [16] and show how the architectural choices made in GOLD can be positioned within the space of dynamic coalitions outlined above. We illustrate our approach to leveraging the benefit of the formal model by describing tools for analysing access control that are based on our modelling work.

The GOLD project is developing a software architecture to support the formation, operation and termination of business coalitions (Virtual Organisations) in the high-value chemicals industry. The production of a chemical involves a large number of stages, including initial experiments, building and running of industrial-scale plant and safety analysis of by-products. A coalition of companies often forms around the production of a single chemical, since few companies have the resources to see

one chemical right through from inception to marketing. These coalitions are loosely bound together with members joining as necessary and leaving when their part of the process is complete. The GOLD architecture uses web services to allow companies to communicate information, transfer documents and access one another's resources. It is intended that each company in the GOLD environment will offer a standard set of services in a uniform way, allowing coalitions to form and operate at a much greater rate than is currently possible.

In the GOLD environment, overlapping coalitions can be formed. In a sense, this environment forms a single global coalition, whose purpose is to form smaller and more constrained coalitions around particular development processes. A single company may be in many such coalitions. We have developed a model of the GOLD architecture, identifying the subspace of dynamic coalitions which it may support. With this in mind, we have begun an ongoing process of review- and scenario-based validation of our GOLD model with domain experts, dealing with each dimension and enhancing the GOLD model in accordance with feedback. Even in early stages, the modelling exercise has identified issues in the GOLD architecture and, where appropriate, these are identified below.

Coalition Membership: Both companies (*agents*) and company employees (*users*) may be members of a GOLD coalition. They may both have associated *roles*.

$$\begin{aligned}\Sigma_{gold} :: \quad & coals : Cid \xrightarrow{m} Coalition \\ & agents : Aid \xrightarrow{m} Agent \\ & users : Uid \xrightarrow{m} User\end{aligned}$$

$$\begin{aligned}Coalition :: \; & members : (Aid \mid Uid)\text{-}\mathbf{set} \\ & aroles : Aid \xrightarrow{m} aRole\text{-}\mathbf{set} \\ & uroles : Uid \xrightarrow{m} uRole\text{-}\mathbf{set}\end{aligned}$$

$$\mathbf{inv}\ (\text{-}, aroles, \text{-}) \triangleq \exists!\, aid \in \mathbf{dom}\, aroles \cdot \text{LEADER} \in aroles(aid)$$

$$Agent :: employees : Uid\text{-}\mathbf{set}$$

The information about a company includes its employees. We must also ask the question "Must a user in a coalition be an employee of a member company?" This is not stipulated by [16], but if in a particular coalition the answer is yes, it can be enforced within the model by an invariant.

Every GOLD coalition will be initiated by a single leader, and we enforce this by the invariant. Membership of the coalition will be at the discretion of this leader, but companies will be able to leave unilaterally. Validation of the model has raised the issue of the procedures surrounding the removal of a leader from a coalition (this leads to a potential invariant violation).

Information Storage: Within GOLD, a *document* "is the fundamental unit of information exchanged between VO members" [16]. Each project will have a set of documents associated with it. These may be stored centrally or on a VO member's local site. GOLD will "provide seamless access to information regardless of its physical location" [16].

An environment-wide registry defines a common understanding of the classes of document which may be exchanged between companies. Agents and coalitions may have *Document Repositories (dr)*. Documents placed in these storage facilities may be retrieved using an index or identifier. This makes a map between identifiers and documents the most natural representation of information storage. An agent in multiple coalitions must keep track of which of their documents are part of which coalition (*coalinfo*).

$$\begin{aligned}\Sigma_{gold} :: \quad & \cdots : \cdots \\ & doc\text{-}type : \text{LAB-RESULTS} \mid \text{MNGT-REP} \mid \ldots\end{aligned}$$

$$\begin{aligned}Coalition :: \; & \cdots : \cdots \\ & dr : Did \xrightarrow{m} Document\end{aligned}$$

$$\begin{aligned}Agent :: \; & employees : Uid\text{-}\mathbf{set} \\ & dr : Did \xrightarrow{m} Document \\ & coalinfo : Cid \xrightarrow{m} Did\text{-}\mathbf{set}\end{aligned}$$

Information Insertion: A GOLD VO will run a project to an agreed project plan. This plan will include which documents should be created and when, and the project member (*User* or *Agent*) tasked with creating them. The creation of a document acts as a milestone for the project plan. When a document is created it may be added to the coalition repository. In the operation below, the user *u* from company *comp* adds a single document *d* to a coalition.

$$\begin{aligned}&AddToVO\ (u{:}\, Uid, comp{:}\, Aid, d{:}\, Document, \\ &\qquad did{:}\, Did, c{:}\, Cid) \\ &\mathbf{ext\ wr}\ coals : Cid \xrightarrow{m} Coalition \\ &\mathbf{pre}\ u \in \mathbf{dom}\, users \wedge comp \in \mathbf{dom}\, agents \wedge \\ &\quad \{u, comp\} \subseteq coals(c).members \wedge \\ &\quad u \in agents(comp).employees \wedge \\ &\quad did \notin \mathbf{dom}\, coals(c).dr \wedge c \in \mathbf{dom}\, coals \wedge \\ &\quad \exists a \in \mathbf{dom}\, agents \cdot a \in coals(c).members \wedge \\ &\quad permits(a, u, \{d\}, c) \\ &\mathbf{post}\ coals = \overleftarrow{coals} \dagger \\ &\quad \{c \mapsto \mu(\overleftarrow{coals}(c), dr \mapsto \overleftarrow{coals}(c).dr \cup \{did \mapsto d\})\}\end{aligned}$$

Within coalition *c*, the project plan will need to be consulted to determine if user *u* is permitted to create that document. The predicate *permits* captures this.

When a document is added, a new access control policy is created for that document. Initially, it will contain rules imposed by the high-level project policy. For example, they may ensure conformance to a legal obligation. When a new document is added to the project the author is entitled to add further access rules to these pre-existing rules.

Information Transfer: An agent in a coalition may request documents from the coalition repository directly, or from another participating agent or user. In both cases the coalition access control policy must be enforced.

The operation below describes the formal transfer of a document from one agent to another. The behaviour of the access control policy is captured as the predicate *authorises*.

A transfer is allowed only if it is authorised by an appropriate member of the coalition. It is important that we identify the particular coalition within which the transfer is taking place. This is because many coalitions may exist at any one time and confidential information relevant to one coalition should not be passed under the auspices of another.

$$
\begin{array}{l}
\mathit{InfoTransfer}\ (\mathit{from}, \mathit{to}\colon \mathit{Aid}, d\colon \mathit{Document}, \\
\qquad\qquad \mathit{did}\colon \mathit{Did}, c\colon \mathit{Cid}) \\
\textbf{ext wr}\ \mathit{agent}\ :\ \mathit{Aid} \xrightarrow{m} \mathit{Agent} \\
\textbf{pre}\ \{\mathit{from}, \mathit{to}\} \subseteq \textbf{dom}\ \mathit{agents} \wedge \\
\quad d \in \textbf{rng}\ \mathit{agents}(\mathit{from}).\mathit{dr} \wedge \\
\quad \mathit{did} \notin \textbf{dom}\ \mathit{coals}(c).\mathit{dr} \wedge \\
\quad c \in \textbf{dom}\ \mathit{coals} \wedge \{\mathit{from}, \mathit{to}\} \subseteq \mathit{coals}(c).\mathit{members} \wedge \\
\quad \exists a \in \textbf{dom}\ \mathit{agents} \cdot a \in \mathit{coals}(c).\mathit{members} \wedge \\
\qquad \mathit{authorises}(a, \mathit{from}, \mathit{to}, \{d\}, c) \\
\textbf{post}\ \mathit{agents} = \overleftarrow{\mathit{agents}} \dagger \\
\quad \{\mathit{to} \mapsto \mu(\overleftarrow{\mathit{agents}}(\mathit{to}), \\
\qquad \mathit{dr} \mapsto \overleftarrow{\mathit{agents}}(\mathit{to}).\mathit{dr}\ \cup\ \{\mathit{did} \mapsto d\})\}
\end{array}
$$

Note that the predicate *authorises* need not require the intervention of an authorising agent for every information transfer: permission to distribute documents may be granted in advance and stored until referenced by the *authorises* predicate.

Other Dimensions: Concerning *governance*, every GOLD coalition will be initiated by a single leader, and we can enforce this by the invariant in Σ_{gold}. The authorisation structure of a GOLD coalition will therefore naturally be a star formation, with the leader as the nexus. Validation of the model has raised the question of how to handle the case of a subcontractor choosing to further subcontract their work. Whether or not this should be treated as a separate coalition is unclear in GOLD.

For *provenance*, the GOLD architecture will provide a secure *archival* service for future audit, available only to authorised members. It can be modelled by creating a document repository at the coalition level. The *InfoTransfer* operation definition is then expanded to include the automatic updating of the archive facility.

The GOLD architecture does not impose a model of trust. Companies will be free to trust (or mis-trust) information, communications and other companies as they choose.

Leveraging the model: We aim to leverage the benefit of our modelling work mainly through "lightweight" formal methods mechanisms, particularly via execution-based analysis of the formal models. This approach is particularly strongly supported by VDMTools[4] which includes an interpreter implementing an operational semantics of VDM-SL. This permits rapid validation of the model using test scenarios, supported by test coverage analysis tools. Further, an application programming interface allows models to be executed in the interpreter via interfaces designed to make the model accessible to domain-based experts. Using this technology, we are developing a workbench to enable system architects to design access control policies for GOLD dynamic coalitions.

A GOLD access control policy is potentially complicated. It may be written by a number of different people, and conflicts and ambiguities may arise. Policies are written in the eXtended Access Control Markup Language [17]. We have developed a model of the semantics of XACML policies in VDM, allowing specific policies to be translated to VDM [18]. A policy writer who adds new rules to the pre-existing policy may confirm that the combined rules behave as anticipated before the policy is updated. Together with the VDM model presented in this paper, this gives us the opportunity to investigate the behaviour of proposed VOs and projects before they are formed. Access requests derived from workflows to be supported by the coalition are translated to calls to the *authorises* function in the *InfoTransfer* operation precondition. These calls are evaluated in the interpreter against the access control model derived from the XACML policy.

Our interpreter-based approach allows coalition developers to analyse new and updated access control policies for weaknesses and inconsistencies before committing them to the running coalition systems. In future, it will allow the developers of other coalitions to ensure that dimensions such as information transfer are described strongly enough to support the workflow requirements.

VII. Conclusions and Future Work

We have used a formal model-oriented specification language as the basis for an exploration of the space of dynamic coalitions. The modelling language's emphasis on abstractions of data, state and operations has encouraged a focus on the "meta-information" that characterises the structure and information flows of a coalition. As a result, several dimensions have been identified along which dynamic coalition structures may vary. We have placed one real virtual organisation scheme, that of the GOLD project, in the space spanned by the dimensions that we have identified. Doing so has identified several areas in which the GOLD architecture can be clarified or improved. Future work is envisaged in several areas, outlined below.

Designing Dynamic Coalitions: The challenge in building systems to support dynamic coalitions lies in providing just sufficient structure to permit validation of emergent properties without over-constraining heterogeneity and flexibility. Possible directions for further work include exploring/validating the dimensions by applying them to a wider range of known and possibly new dynamic coalition structures. A particular area of interest is in predicting the consequences for information flow of two coalitions merging. We also intend to investigate the description of further structures that are superimposed on the coalition itself and investigate policy languages for describing access control on the basis of meta-information.

Although the work done so far has been within a formal modelling framework, many of the potential benefits do not

[4] `http://www.vdmbook.com/tools.php`

require the designer to use formal apparatus. For example, we intend to develop proof obligations for dynamic coalitions but have also begun to use the formal models to develop checklists for coalition architects working in each dimension.

Validation of Dynamic Coalition Models: Discharging consistency obligations and domain-specific validation conjectures for VDM models can be done at various levels of confidence, ranging from testing to formal proof. For models of dynamic coalitions, one may envisage several ways of leveraging further benefits from exposing the model to domain experts. An executable model linked to a suitable interface permits ad hoc exploration of the coalition model by a user [9], [10]). With features such as invariant and precondition checking, this permits systematic analysis of normative as well as some failure behaviours. Scenario scripts can be defined and executed, representing paths through state transition models. Indeed, under strict constraints, it may be possible to search systematically for states having specific properties.

Domain Based Security: Domain Based Security [19] focuses on the way information is shared. We would like to integrate this model with our existing models of dynamic coalitions, in order to examine and predict how dynamic coalitions might function in a Domain Based Security environment.

Responsibility: Section II explores the question: *where does responsibility lie for coalition membership?* We want to broaden this question to include all coalition actions. Some way of "keeping records" of coalition behaviour may be valuable, so that these questions can be asked retrospectively, and that a basis can be provided for recovery from undesirable states.

Acknowledgments: We are grateful to Tom McCutcheon and Ramsay Taylor of the UK Defence Science and Technology Laboratory (DSTL) for encouragement to examine information flow in dynamic coalitions. We acknowledge much helpful input from colleagues in the Interdisciplinary Collaboration on Dependability of Computer-based Systems, especially Peter Ryan, Michael Harrison and Fred Schneider, as well as our VDM colleagues Peter Gorm Larsen and Marcel Verhoef. We are grateful to the referees for helpful comments guiding the revision of the paper. This research was part funded by both the DSTL and the UK EPSRC under the GOLD e-Science Pilot Project.

REFERENCES

[1] N. Sanchez, D. Zubiaga, J. González, and A. Molina, "Virtual Breeding Environment: A First Approach to Understanding Working and Sharing Principles," in *Proceedings of InterOp-ESA'05*. Springer, 2005.

[2] P. G. Larsen, "Coalition c2 interoperability challenges," in *Proc. 11th. Intl. Command and Control Research and Technology Symposium, Cambridge, UK*. DOD Command and Control Research Program, US, 2006, to appear at `http://www.hsdl.org`.

[3] N. Lethbridge, "An I-based Taxonomy of Virtual Organisations and the Implications for Effective Management," *Informing Science*, vol. 4, no. 1, pp. 17–24, 2001.

[4] R. Klüber, "A framework for virtual organising," in *Proceedings of the VoNet Workshop*, 1998, `http://www.ve-forum.org`.

[5] H. Khurana, V. Gligor, and J. Linn, "Reasoning about joint administration of access policies for coalition resources," in *Proceedings of IEEE Int. Conf. On Distr. Computing (ICDCS)*, 2002.

[6] V. Bharadwaj and J. Baras, "Dynamic adaptation of access control policies," in *Proceedings of Military Communications Conference (MILCOM 2003)*, 2003.

[7] D. Allsopp, P. Beautement, J. Bradshaw, E. Durfee, M. Kirton, C. Knoblock, N. Suri, A. Tate, and C. Thompson, "Coalition agents experiment: Multi-agent co-operation in an international coalition setting," *IEEE Intelligent Agents*, vol. 17, no. 3, pp. 26–35, May-June 2002.

[8] C. B. Jones, *Systematic Software Development using VDM*, 2nd ed. Prentice Hall International, 1990.

[9] J. S. Fitzgerald and P. G. Larsen, *Modelling systems: practical tools and techniques in software development*. Cambridge University Press, 1998.

[10] J. S. Fitzgerald, P. G. Larsen, P. Mukherjee, N. Plat, and M. Verhoef, *Validated Designs for Object-oriented Systems*. Springer-Verlag, 2005.

[11] I. J. Hayes, Ed., *Specification Case Studies*, 2nd ed. Prentice Hall International, 1993.

[12] J.-R. Abrial, *The B-Book: Assigning programs to meanings*. Cambridge University Press, 1996.

[13] P. G. Larsen, J. S. Fitzgerald, and T. Brookes, "Applying Formal Specification in Industry," *IEEE Software*, vol. 13, no. 3, pp. 48–56, May 1996.

[14] J. W. Bryans, J. S. Fitzgerald, C. B. Jones, and I. Mozolevsky, "Dimensions of dynamic coalitions," School of Computing Science, Newcastle University, Tech. Rep. CS-TR-963, May 2006.

[15] D. J. Andrews, Ed., *Information technology – Programming languages, their environments and system software interfaces – Vienna Development Method – Specification Language – Part 1: Base language*. International Organization for Standardization, December 1996, International Standard ISO/IEC 13817-1.

[16] A. Conlin, N. Cook, H. Hilden, P. Periorellis, and R. Smith, "GOLD Architecture Document," School of Computing Science, Newcastle University, Tech. Rep. CS-TR-923, 2005.

[17] OASIS, "eXtensible Access Control Markup Language (XACML) version 2.0," `http://docs.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml/`, OASIS, Tech. Rep., Feb 2005.

[18] J. W. Bryans, J. S. Fitzgerald, and P. Periorellis, "Model based analysis and validation of access control polices," School of Computing Science, Newcastle University, Tech. Rep. CS-TR-976, July 2006.

[19] K. J. Hughes, "Domain Based Security: enabling security at the level of applications and business processes," White Paper, QinetiQ, 2002.