# Contract Performance Assessment for Secure and Dynamic Virtual Collaborations

Theo Dimitrakos[1*], Ivan Djordjevic[2*], Zoran Milosevic[3], Audun Jøsang[3], Chris I. Phillips[2]
[1]Central Laboratory of the Research Councils, Rutherford Appleton Laboratory, OX11 0QX, UK
[2]Queen Mary University of London, Mile End Road, London E1 4NS, UK
[3]Distributed Systems Technology Centre, Brisbane, Australia

## Abstract

*In this paper we sketch a framework supporting contract enactment within the context of virtual organisation units that are dynamically created in order to achieve a common objective by securely sharing resources, services and information. The framework is built on top of a joint extension of the policy deployment architecture for peer-to- peer communities that we proposed in [1] and the contract enactment capability described in [6] that enables monitoring, mediation, arbitration and enforcement of electronic contracts in multiple, simultaneous closed collaborations. A longer-term goal is to deliver a scalable method of setting up contract enforcement and contract performance management infrastructures for inter-organisational information systems that allow the on-demand creation and dynamic evolution of secure Virtual Organisations based on the ad-hoc integration of systems across Enterprise boundaries.*

## 1 Introduction

The Internet provides a ubiquitous, standards-based substrate for global communications of all kinds. Rapid advances are now being made in agreeing protocols and machine-processible message/document formats that will soon enable open application-application communication and brings about the prospect of *ad hoc* integration of systems across organisational boundaries to support collaborations that may last for a single transaction or evolve dynamically over many years. Effectively, we will witness on-demand creation of *dynamically-evolving, scalable* **Virtual Organisations (VO)** spanning national and enterprise borders, where the participating entities pool resources, capabilities and information to achieve common objectives.

This paper deals with the problem of supporting secure, trusted and predictable interactions between parties involved in a VO throughout its life-cycle. By Virtual Organisation we mean a dynamically created collaborative arrangement between existing organisations or their organisational units to achieve a specific objective. Typically, the life cycle of a VO is shorter than life-cycle of traditional organisations.

This paper proposes the use of **Closed Collaboration Teams (CCT)** paradigm that is a generalisation of the Closed User Group architecture discussed in [1], to support secure and trusted interactions in VOs. CCT provides a dynamic and distributed support for **peer-to-peer (P2P)** collaboration, taking into account hierarchical administration requirements. It allows a scalable method of setting up security infrastructures that has the benefits of allowing P2P collaboration, whilst maintaining the robustness and re-configurability of systems supplied by the central administration of the security policies.

Secure and trusted operations of VOs represent one prerequisite for their functioning. A further requirement is to support the interactions between VO's constituents in a similar way as in traditional organisations to ensure that their interactions comply with their business agreements. To this end the VOs can make use of services provided by the specialised contract management systems. These services for example facilitate recording of their newly created terms of agreement, monitoring of their performance and notifications and enforcement. The contract management system used to support the VOs requirements is based on the **Business Contract Architecture (BCA)** paradigm described in [6], [7], [8].

The main novelty of this paper is an integration of CCT (as a flexible and trusted distributed collaborative architecture) and BCA (as an enterprise wide distributed contract management architecture). The primary aim of this integration is to support secure, trusted and agreement-complaint functioning of VOs but also other forms of extended enterprises.

The paper is structured as follows. Section 2 present key ideas behind the CCTs. Section 3 outlines main architectural components of BCA. In Section 4 we consider integration of these two architectures. Conclusions and Future Work are outlined in section 5.

## 2 Closed Collaboration Teams

CCTs are inspired by variant of the **Closed User Group (CUG)** concept proposed in [3] and brought further in [1] for multi-domain security management in

[*] Corresponding editors: theo.dimitrakos@rl.ac.uk at CLRC and ivan.djordjevic@elec.qmul.ac.uk at QMUL

virtual organisations. They provide a new distributed and secure dynamic environment for collaborative working without topological constraints.

In its simplest form the CCT model distinguishes two main classes of roles: *team members* and *team manage*r.

- **Team members** are peers who are informed of each others identity and location and are interacting with each other on a peer-to-peer basis by using of some type of group certificates embedded in the messages they interchange. Messages with certificate information that is not matching the required team certificate are either deleted or ignored without further processing.
- **Team manager** is responsible for managing team membership and issuing or updating team certificates. Team manager of one level in the organizational hierarchy may themselves be viewed as members of a CCT at a level above the level of the teams they manage.

In the context of this paper, we assume for simplicity that each entity in an organisation is primarily assigned to a single manager called its ***"local" manage***r. Clearly, each manager can have many "local" members. This structure resembles line management in structured organisations. In the context of this paper, we treat the restriction to a unique "local" manager as a convention. However, it is worth noting that, certain network topologies impose such locals as their primary topological structure (e.g. users of a LAN and LAN administrator).

All existing CCT members are informed of the arrival or departure of a client member and the CCT exists until the last member leaves, be it the creator or any of the members. Interactions between CCT peers are assumed to be based on exchange of certified messages. For the purpose of this paper, we distinguish two types of certificate information:

- ***me2me:*** member-to-member certificate information is unique to each team, and is used to enable direct peer-to-peer communication between the members of CCT, such as file transfers, process invocation (via Remote Procedure Calls), etc.
- ***me2ma***: member-to-management certificate information is unique to each team manager and enable interaction related to CCT management. (It is assumed that a single manager is responsible for each specific CCT).

## 2.1Virtual Closed Collaboration Teams

Members of CCT teams can initiate or join "virtual" CCTs which span across organisational units. A virtual CCT can be initiated by any member of an existing CCT and any member of an existing CCT can request to join a virtual CCT[1]. Virtual CCTs are assigned a ***"remote" manager*** chosen (e.g. via self-initiative or voting) among the pre-existing CCT managers which are contributing members to the virtual CCTs. For any member of an existing CCT, say X, to be able to initiate the creation of, or joining to, a "virtual" CCT, the prior endorsement of its "local" manager is required. Depending on the virtual CCT policy, joining a CCT may also require endorsement of the "remote" manager and/or the management of each local that is contributing member to the virtual CCT.

Based on its objective, clients may be free to decide to participate in a CCT of their interest. Also, participation of a client could be compulsory, based on a focus of the CCT and purpose of CCT creation (e.g., within a company/organisation, different resources and/or employees could be allocated for a specific project/work task that forms a basis for CCT creation). Once allocated, the responsibility of CCT management and maintenance remains with the "virtual" manager until CCT termination. Managers are responsible for managing and maintaining the CCT policy. CCT membership is determined by the possession of the appropriate certificates, which are endorsed and recognised by the CCT management. Only members possessing an appropriate certificate can participate in the CCT, while new members receive a CCT certificate that defines the set of privileges (group role) of each particular member. As we explain in subsection 2.3, CCT may rely on external certificate authorities (CA) in order to provide the initial identities.

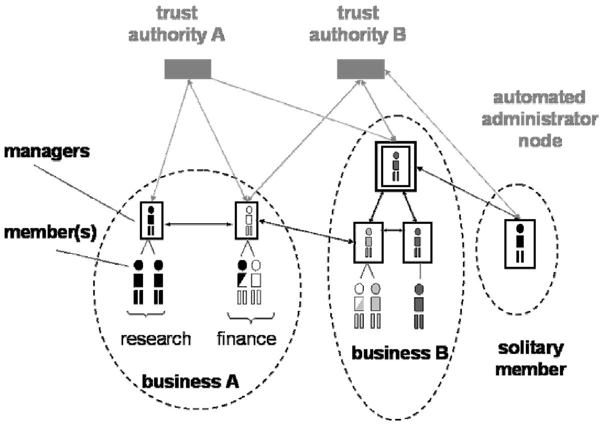

Figure 1: Basic CCT Structure

## 2.2 Basic Interactions in Virtual CCTs

Figure 2 provides an overview of the dynamic introduction of a new member X in a virtual CCT named V, that is managed by R who is primarily responsible for CCT local B. X originates from an organisationally distinct CCT local A, managed by L.

[1] We assume that CCT managers may publish information about existing CCTs and any CCT members can discover and inspect information about existing CCTs, using for example similar mechanisms to the Web Service publication, inspection and discovery.

Member X is prevented from contacting directly R whose virtual team V it wishes to join. Instead X sends messages to R routed via its local manager L, who has also to endorse X's intention to join the virtual team V. If L approves, then L effectively acts as a "proxy" member and attempts to "join" the team V on X's behalf. If R accepts this, then L will receive a certificate for **me2me** communication within the scope of team V, which L will forward to X. By possessing this certificate, X is able to participate in direct **me2me** interaction within the scope of team V. Notably, L retains a degree of control over its client: First, from X's perspective all **me2ma** communication is routed via L, who is able to endorse and has to countersign the messages. Second, from R's perspective L is the first recipient of all communication intended to X. Notably, L would be the intermediary for any number of members of team A participating in the virtual team V, maintaining the interactions between R and each A team member distinct.

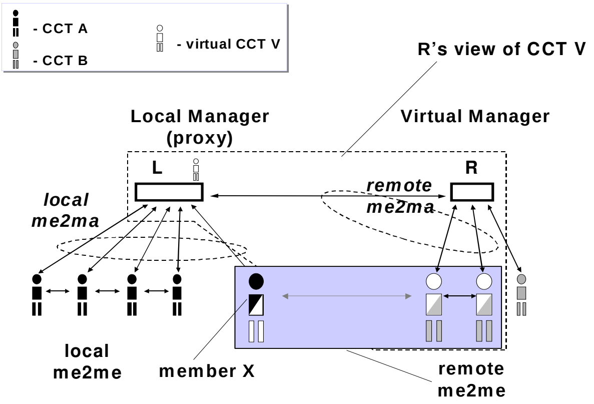

Figure 2: Local Manager / Proxy Member Behaviour

## 2.3 Digital Identity Management Considerations

The previous sections have described CCTs with the assumption that pre-existing member identities are readily available. A fundamental question for dynamic virtual collaboration is whether an identity can be trusted. More specifically this means that an entity's online identity should be fixed and that it can be mapped to the entity's real world presence. This section describes principles for establishing reliable identities within our framework.

The so-called Web-PKI consists of many isolated hierarchic PKIs where the root public key of each isolated PKI is hard-coded in the distribution software of all the major Web browsers. Each isolated PKI thus represents a closed group with a unique name space. Uniqueness of member names is not necessarily guaranteed across different PKIs unless the name is an Internet Domain Name, which per definition is unique. Despite this potential problem, Web servers and clients can have their certificates from different PKIs, and still be able to authenticate and communicate securely with each other.

We propose to base CCT members' identities on public-key certificates issued by **Certification Authorities (CA)** outside the CCT. Members can choose any commercial CA belonging to any PKI as long as the PKI is recognised by the CCT Managers. This requires the CCT managers to obtain multiple PKI root public keys in order to recognise certificates issued by CAs belonging to different PKIs. This is similar to the way Web-browsers are able to recognise server certificates issued by different CAs because the root public key of each isolated PKIs is hard-coded in the browser software.

External CAs provide the required physical infrastructure that is required. By relying in external commercial CAs, CCTs will be able to issue certificates without the need for any additional physical infrastructure to their standard Information and Communication Technology. This is consistent with the intention to the have CCTs operate online as far as possible.

In a typical scenario party X and party Y contact manager M, or are invited by manager M to join a particular virtual CCT. Party X presents a public-key identity certificate issued by a CA in PKI$_1$ and party Y presents an identity certificate issued by another CA in PKI$_2$. Manager M recognises both PKIs and is able to validate both certificates because M has already obtained the root public keys of PKI$_1$ and PKI$_2$. Party A and B will then be issued with me2me and me2ma certificates to be used within the virtual CCT. This is illustrated in Figure 3 below.

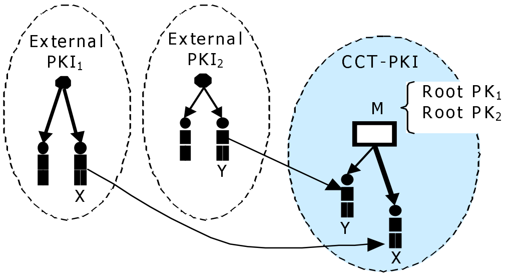

Figure 3: External PKIs and CCT-PKI

The certificate structure of each CCT can be seen as a single-level autonomous PKI that has been established for the purpose of the CCT only and where the CCT manager represents the CA. If anonymity is required by a member, the manager can issue CCT certificates with fictive member identities. Accountability can still be assured by the manager being able to translate the anonymous identity to the real identity if required e.g. in case of dispute resolution.

## 2.4 CCTs seen as flexible virtual firewalls

The CCT interaction model allows for hierarchies of communities of peers, where peers at the same level of an organisational structure may interact with each other on a P2P basis and with their local manager within closed VO teams. Local managers intercept communication between peers across different levels of the organisational structure within the same or different enterprises. This is combined with a basic security enforcement that provides the functionality a distributed firewall software application. For the purposes of the architecture described in this paper, we can assume a simple **Software Firewall Application (SFA)** residing[2] with each of the entities in a CCT local (including the local manager). SFA can be understood as an enforcement agent enacting in-bound / out-bound communication policy controls in a preventive way. That is, each CCT member's SFA executes a set of rules, defined and owned by the local CCT manager and relating to the role of each CCT member. These rules regulate in-bound and out-bound communication. For example, the SFA of a peer M blocks all messages addressed to M that do not have a ***me2ma*** certificate or a ***me2ma*** certificate relating to some CCT that M is a member of. To do this SFA may compare against a local list of certificates accepted by *M*. Blocking out-bound communication is potentially more demanding (although often less critical). In the case of ***me2me*** communication, for example, in addition to the ***me2me*** certificate, SFA has also to verify that the intended recipient is a member of the CCT that the ***me2me*** certificate relates to. As we explain in [1], preventive enforcement of more refined rules relating to permission and prohibition policies can also be supported by enabling SFA to interpret and execute enforcement rules interpreting such policies and incorporating essential information about the privileges of a peer in m2m certificates.

The CCT model described above addresses one aspect of collaborative interactions in VOs – dealing mostly with permission and prohibition policies. However, rules of engagement in VOs also include the expression of obligation policies and require an additional framework for checking fulfilment of obligations. To this end the CCT model can be augmented with the capabilities of a contract management system such as previously described in [6]. Although more dynamic than conventional distributed firewalls, the CCT enforcement model has to become more flexible in order to secure the dynamic VO configurations facilitated by the CCT management model. For example, members of a virtual CCT A can indirectly compromise the operation of other competing virtual CCTs by causing damage to their shared resources through apparently legitimate interactions in the context of A. One way to alleviate such interference is by introducing CCT contracts that:

– Necessitate partitioning resources dedicated to different CCTs;
– Require that, when a member wishes to join a virtual CCT, its local manager should inform all of its existing remote managers about the local member's intentions and that the local manager should seek prior agreement of each exiting remote manager before allowing its local member to join that virtual CCT;
– Take advantage of a capability enabling collaborative contract performance monitoring and notification among CCTs;
– Take advantage of collaborative contract enforcement mechanisms across CCTs.

The above provide additional motivation for enhancing the CCT concept through the introduction of an effective trust and contract management capability, which is main focus of the rest of the paper.

## 3 Contract Establishment and Execution

This section describes the basic components of architecture for contract establishment and execution based on [6]. The remainder of the section depicts this role-based architecture, and indicates key information flow between the roles which are in general involved in more than one process. The architecture represents an extension of the BCA described in [7].

### 3.1 Roles Supporting Contract Establishment

The following roles support the process of establishing a contract:

– **Negotiator** mediates the negotiation process. During the negotiation phase, parties can exchange offers and counter-offers containing one or more of the following: contract templates, individual contract clauses and finally contract variables that are negotiable items. During contract negotiations it may be possible to perform certain aspects contract validity checking as mentioned below.
– **Validator** ensures the creation of legally valid contract instances, assessing proposed contracts against various aspects of contract validity such as competence, clarity, legal purpose and consideration elements. See [8] for further details on contract validation.
– **Notary** is a trusted party that stores contract instances after the contract has been agreed upon, checked for validity and signed by both parties. Such contract instances can be later used as evidence of agreement in the contract monitoring and enforcement activities. Also, notary component can be hosted by one or both parties involved in contract.
– **Contract Forms Repository** provides storage and access to standard contract forms or contract clauses, depending on contractual scenario. It can be used by

[2] SFA may physically reside with the CCT member or act as a proxy associated with it. In the context of this paper we do not distinguish between these two options.

parties to the contract who use pre-defined contract forms to produce individual contract instances or by contract drafters who are defining building blocks for contracts. There may be also a need for a specialised contract templates editor that can provide functionality of both text editing but also type definitions for the fields that represent negotiable items within the contract.

### 3.2 Roles Supporting Contract Execution

The following roles support contract enforcement and performance monitoring during the performance of a contract (Figure 4).

- **Monitor** enables monitoring of the activities of parties, measuring their performance if needed and recording the relevant events. It can also signal a contract non-performance to the Discretionary Enforcement Moderator (see below) if it detects such an event.
- **Notifier** implements various notifications mechanisms needed to send warning messages to indicate a pending contract-significant event, including possible non-compliance event that may be detected. To simplify presentation, Notifier is not shown on the Figure 4.
- **Enforcer** applies enforcing actions directly to the parties to ensure that some specific behaviour conforms to the contract. From a control theory point of view, this role is analogous to an actuator.
- **Discretionary Enforcement Moderator (DEM)** forms an opinion about the extent of deviation by the non-performing parties. Once the arbitrator forms such an opinion, it chooses a route of action which may invoke settlement leading to the success of a suitably amended transaction. Alternatively, it may endorse the enforcement of corrective measures to be executed by a preventive security mechanism realised by the Contract Enforcer role. (An overview of the DEM's decision making procedure is modelled as a finite state machine in [6].) The DEM forms its opinions on the basis of evidence about deviation of the non-performing parties, that is provided by the Contract Monitor, external advisors and possibly additional recommendations from agents representing the parties, in a spirit similar to a (human) judge's process for arriving at his ruling. During this process the DEM component may take the following specific roles.
- **Mediator** - who initiates a settlement leading to the success of an amended transaction or decides failure of mediation leading to the invocation of arbitration.
- **Arbitrator** - takes over when a settlement as per above cannot be reached, or when a party's deviation from the expected performance is high enough to justify the deployment of corrective measures. An arbitrator may initiate the enforcement of corrective measures through the Contract Enforcer, leading to the recoverable failure of the transaction and, potentially to penalising the non-performing party. In the absence of any suitable corrective measures, the Arbitrator may signal correction failure, in which case the Contract Validator is informed so as to prevent further access to the system by the non-performing parties, if necessary, and the case is carried on outside the Contract Architecture.

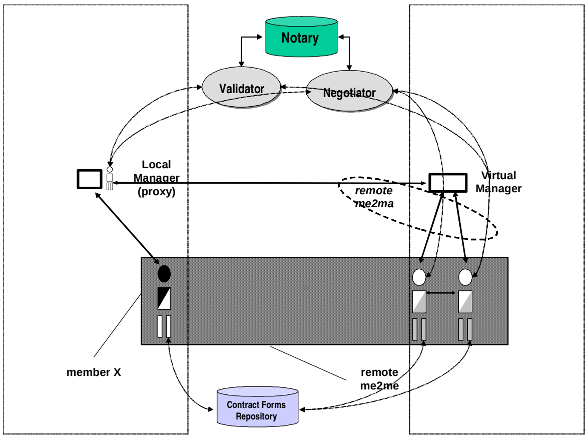

Figure 4: Basic contract establishment roles across CCT

# 4 Method Integration

In this section we discuss an integration of the above role-based architectures for CCTs and Contract Establishment and Executions. The result of this fusion is a novel architecture enabling the different organisational units to negotiate their terms of involvement in virtual organisations, encode them within electronic contracts and use this as a basis for governing their interactions. This governance is facilitated by using a distributed inter-organisational contract management infrastructure ensuring predictable, contract-complaint interactions – and augmented with the secure and trusted interactions in accordance with the policies stated in the CCT.

Each virtual organisation unit amounts to a virtual CCT which is executing a distinct contract. Each CCT member may participate in any number of contracts, characterised by the number of the virtual CCTs in which they participate. Different CCTs share distributed capabilities for contract monitoring and enforcement.

A motivation determining the choice of integration is on the one hand, to improve the manageability and security of CCTs by sharing monitoring and enforcement mechanisms between them, while on the other hand, to secure the execution of each contract by ensuring that members who are not engaged in the contract are prevented from directly interfering with the contract execution. The integrated architecture is presented in the following subsections.

## 4.1 Contract establishment

CCT locals can be understood as a common abstraction of various different types of moderated closed communities. Different types of local policies may be used to govern such communities depending on their objective. Often the local policies would be an extension of some enterprise-wide policies that apply to each CCT local, as an organisational unit within an enterprise. To realise secure collaborations across CCT locals, however, entities will need to form virtual CCTs, which are governed by some collaboration agreement, encoded by means of an electronic contract. The formation of such collaboration agreements requires that the prospective virtual CCT members negotiate their terms of involvement.

The contract governing the operation of a virtual CCT may also refer to the policies of the participating CCT locals, and the conditions in the contract will effectively determine the role of the signatory members in the CCT, their privileges and obligations, as well as any sanctions that may automatically apply in basic cases of non-performance.

The process of negotiating a contract with a virtual CCT becomes more efficient in the presence of supporting services implementing the roles described in section 3.1. Once a virtual CCT is initiated the prospective virtual CCT manager is responsible for initiating the preparation of a contract template. Depending on contractual scenario, a Contract Forms Repository is used to store standard contract forms or contract clauses, to be used by the CCT in order to produce the specific contract instance governing its operation. The contract negotiation process is moderated by the virtual CCT manager and facilitated by the mediation of a contract negotiator. The latter can be either a trusted third party or a functionality implemented collectively by the CCT community – in which case the CCT manager is directly responsible for orchestrating the contract negotiation process.

Typically, contract negotiation is an iterative process where contract drafts are periodically checked by the contract Validator who ensures the creation of a legally valid contract. In most cases the Validator will be a trusted third party outside of a CCT local. Once the contract instance governing a CCT local is agreed, the Notary stores it. The Notary is a trusted third party outside of a CCT local. Since CCT contract instances may refer to agreements that include references to Industry-wide regulations, one can expect both Industry-wide and enterprise-wide Notaries. The latter maintain validated contract instances about the CCTs locals within an enterprise, which include references to more basic Industry-wide contract instances maintained by the former. Emerging Web technology standards such as xLink (http://www.w3.org/TR/xlink/) can facilitate the management and maintenance of contract instances that are effectively distributed across distinct Notary services.

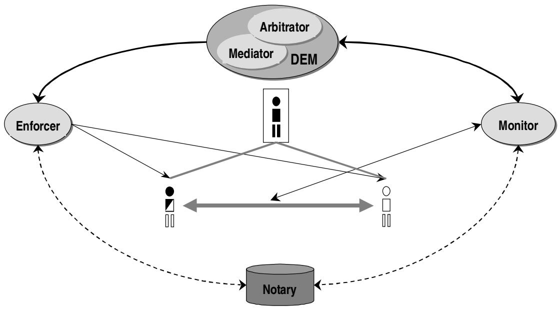

Figure 5: Basic contract execution roles within a CCT local

## 4.2Contract enactment within a CCT local

Contract enactment within a CCT local is supported by a Moderator (DEM) who is capable of arbitrating about the performance of a CCT on the basis of evidence provided by a CCT Monitor. In the case of non-compliance by a CCT member, the Moderator will initiate either a mediation process, managed by a Mediator, or the enforcement of any sanctions applicable, which will be carried out by the Enforcer.

In this subsection we place particular emphasis on performance assessment and enforcement during the enactment of a contract within a CCT local, as the contract monitoring and enforcement management mechanisms of CCT locals also underpin contract enactment in virtual CCTs that span across several CCT locals. We also explain how the main operators of subjective logic [4] provide a formal foundation for encoding, communicating and collating evidence and support arbitration.

**4.2.1 Performance monitoring.** We assume a Monitor capability associated within each CCT local that enables monitoring of the activities of parties, measuring their performance and recording the relevant events. It is subscribed to contract significant events and when these occur, it evaluates the policies for these events, against the agreements that are stored in Notary.

We propose that evidence can be associated to the monitoring of events. Associating confidence to events is reasonable for virtual organisations built on top of dynamic service environments. Such environments, distributed over WAN, typically use asynchronous communication of events that often takes place on a "push" basis: "consumers" subscribe to events and "producers" are obliged to notify them when they produce an event by sending a message of a specific type. In such situations, the occurrence of an event may be uncertain for a variety of reasons including: failure of the “producer” to produce a notification about an event that occurred, delays with the notification, malicious notification about an event (e.g. a failure) that did not occur, etc. In such environments, a CCT Monitor act a as an intermediary facilitating event collection: it collects events generated as a result of CCT interactions and either they forward them to a CCT management capability (e.g. the Moderator) or they analyse them and generate a derived for the CCT management. The derived event may depend on the occurrence of a number of potentially interdependent events within the CCT.

Although we may associate confidence parameters with an event, we do not assume any non-trivial reasoning on decision making to be undertaken by a Monitor. Effectively, monitors report the events they monitor, even when they themselves can only get second hand evidence. The CCT Moderator does the reasoning and makes the decisions. The monitors can supply degrees of confidence with each reported event, so that the CCT Moderator has a basis for reasoning and making decisions. If monitors need to do some complex analysis in order to determine if an event has occurred, e.g. indirectly via related events, then performing that analysis is a separate function which might rely on Subjective Logic or other techniques. (Neural networks could for example be used to detect deviation from typical behaviour, in order to report that an event is suspect.)

Depending on the configuration of the CCT local we distinguish three different kinds of monitoring: *centralise*d, *devolve*d, and *locally coordinate*d. Each kind of monitoring necessitates Monitor behaviours that are operationally different (when viewed from within a CCT local) but observationally similar (when viewed from the outside of a CCT local). We examine each of kind in turn:

– **Centralised.** A centralised monitor is subscribed to contract significant events and when these occur, it evaluates the policies for these events, against the agreements that are stored in Notary. It also passes the results of the evaluation to other components as needed. In relation to the local **me2me** communication, the functionality of the monitor is restricted to observing local events, monitoring network traffic and occasionally intercepting messages sent or received by CCT members. Should events created within a CCT local need to be communicated outside the local, Monitor takes the role of the intermediary in such communication.

– **Devolved.** A devolved monitor can be effectively understood as an abstraction representing a collective realisation of a monitoring capability. That is, each CCT member comes with its own "atomic" (in relative terms) monitoring capability, which contributes to the formation of a collective opinion regarding a potential deviation from the agreed level of quality. Devolved monitoring appears to be a natural choice of monitoring scheme for CCTs that are formed in the absence of uniform and sophisticated underlying infrastructure management services. For example, groups of handheld devices forming ad-hoc networks via point-to-point infrared, satellite or wireless links.

– **Locally Coordinated.** A locally coordinated Monitor combines the behaviour of devolved monitoring with a centralised coordinator who uses a *trust metric* referring to the competence of each CCT local member for the monitoring task and weights the evidence provided by each local member against trust in its monitoring capability. Each member is contributing evidence encoded in a message, which is a (potentially distinguished) part of its **me2ma** communication and can be encoded in a special **me2ma** certificate.

### 4.2.2 Contract Enforcement.

We assume a Moderator and an Enforcer capabilities associated with each CCT local that collectively enable imperative or discretionary enforcement during contract enactment. By *"discretionary enforcement"* we refer to contract enforcement actions aiming at alleviating or correcting performance deviations through mediation (in order to reconcile non-performing CCT members and initiate a settlement leading the success of an amended transaction) or through arbitration by either deploying corrective measures or by referring settlement to an authority outside the CCT structure while "freezing" or "revoking" CCT membership of non-performers. By *"imperative enforcement"* we refer to contract enforcement actions that are performed in a non-discretionary way and directly to the parties in order to ensure that some specific behaviour conforms to the contract. For example, to invoke some access control enforcement mechanism in order to prevent access to a CCT resource, responding to a potential intruder, or revoking the membership of a misbehaving party.

In general, the Moderator orchestrates enforcement by delegating proactive enforcement actions to Enforcer, and discretionary enforcement actions to Mediator or to Arbitrator (if mediation fails or the deviation is too serious)[3]. The choice of enforcement route is determined by the Moderator's opinion about the extent of deviation and risk it entails.

Imperative enforcement can be further distinguished in proactive and reactive and this entails a further dichotomy to the functionality captured by the Enforcer. Proactive enforcement actions are either dictated by the contract (or a CCT local policy) in order to prevent a foreseen threat, or the Moderator initiates them in order to avoid critical deviation or costly contract violations. Reactive enforcement actions are either dictated by the contract as sanctions for non-compliance or the Moderator initiates them as a result of arbitration following a failure to settle non-performance through mediation. As indicated in [1], the mechanisms required for performing proactive and reactive enforcement actions are behaviourally different.

[3] See section 3.2 for a summary of the Moderator role. See also [6] for a more elaborate presentation of this part of the contract enforcement model emphasising on discretionary enforcement options.

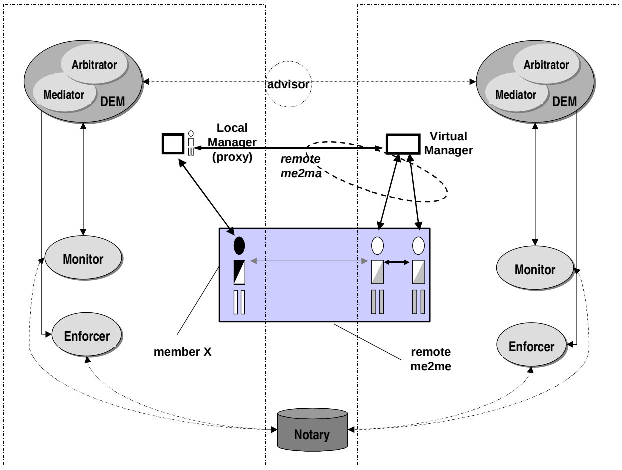

Figure 6: Basic contract execution roles for virtual CCTs

## 4.3Contract enactment across CCT locals

Contract enactment across CCT locals may require collaboration between their corresponding Moderators, Monitors and Enforcer components. The effectiveness of collaborative contract enactment depends on the right distribution of control, liability, and privileges between the main roles participating in the enactment. In particular, the distribution of control strengthens the dependency of collaborative contract enactment on trust relationships between these roles. As in the case of analysing contract enactment within a CCT local, and in order to ease readability, we distinguish performance monitoring from contract enforcement. We also explain how the main operators of Subjective Logic [4] provide a formal foundation for encoding, communicating and collating evidence and support arbitration.

**4.3.1 Performance monitoring.** The Monitor capability of a virtual CCT effectively amounts to a conceptualisation of a network of CCT local Monitors, each of which are responsible for monitoring of the activities of parties, measuring their performance, recording and reporting the relevant events within their CCT local. Each CCT local monitor participating in a virtual CCT monitor is subscribed to contract significant events (within its own locality) and when these occur, it evaluates the policies for these events, against the agreements that are stored in Notary.

The virtual CCT monitor also needs to have the ability to pass the result of evaluation to other components, as needed. These other components can be a Notifier, which simply send notifications formatted in appropriate way to the parties involved or to a Moderator who is capable of further more sophisticated processing such as mediation and arbitration, potentially leading to subsequent mediation or enforcement actions.

Recall that evidence may be associated to events. The virtual CCT will combine the evidence contributed by each local monitor with a *trust metri*c, which refers to confidence in the competence of each CCT local for the monitoring task. In case of performance assessment, dispute, or potential non-compliance, where decision making may be required, this trust metric will be used in order to discount the evidence provided by each CCT local (and communicated via its local Manager) against trust in its monitoring capability.

A Monitor for a virtual CCT can be understood as a network of CCT local Monitors. Each local Monitor is subscribed to contract significant events and when these occur, it evaluates the policies for these events, against the agreements that are stored in Notary. Evidence is then communicated to the virtual CCT via the local management. (Recall that the local Manager manages all administrative communication to and from the outside of the CCT local.) The virtual CCT incorporates the evidence contributed by each local monitor using a *trust metri*c, which refers to the competence of each CCT local for the monitoring task, in order to discount the evidence provided by each CCT local (and communicated via its local Manager) against trust in its monitoring capability.

In subsection 4.3.4, we provide a scheme for incorporating evidence provided from the participating CCT local monitors. Depending on the configuration of the participating locals (i.e. *centralise*d, *devolve*d, and *locally coordinated* examined in subsection 4.2.1), different kinds of monitoring schemes instances may be produced. The scheme is intended for use by the Moderator of the virtual CCT, or any other role who is responsible for decision making. Being analogous to a "network of sensors", the Monitor of a virtual CCT does not perform decision making itself. However, to correctly incorporate the reported evidence into a decision making mechanism requires a scheme that reflects the configuration of such a "network of sensors".

**4.3.2 Contract Enforcement.** Enforcement in a virtual CCT is orchestrated by a Moderator. This can be either a Trusted Third Party or the entity undertaking Moderator's role in the virtual CCT manager's own locality. In any case the authority of the Moderator has to be accepted by every member of the virtual CCT and their corresponding local managers.

In analogy to the case of a CCT local, the Moderator associated with a virtual CCT provides a Mediation and an Arbitration capability, and pending on the opinion it has formed about the extent of deviation by the non-performing parties. In addition to considering evidence from the Monitor and the parties themselves, the Moderator may seek recommendations from some or all the rest of the CCT membership and from external Advisors in order to reduce its own uncertainty or reassess its trust to the parties providing partial evidence. Subsection 4.3.4 elaborates on the use of Subjective Logic for this purpose and subsection 4.4 illustrates our approach by means of an example. Once the Mediator forms such an opinion, it chooses a route of action. Depending on the extent of non-performance, action may invoke settlement leading to the success of a suitably amended transaction, deployment of corrective measures or contract failure.

In [6] we provided an abstract interpretation of the contract enforcement processes as a finite state machine (Figure 7). We used five different levels as means of classifying the states and transactions between the sates of the enforcement process according to the degree with which the transaction execution is on compliance with the prescribed agreement. Level 1 reflect full compliance whereas level 5 reflect total transaction failure and the inability to apply corrective measures to the non-compliant party.

*Level 1* means that the transaction is executed according to the prescribed contract. Deviations can occur as long as notifications to the contractual parties result in the execution getting back on track. *Level 2* means that the transaction has deviated from the prescribed contract, and warnings to non-compliant parties have been ignored. The Monitor informs the DEM which in turn invokes the Mediator, and an attempt to establish an amended contract between the two parties is initiated. In case of settlement the contract execution returns to Level 1 and resumes with the amended contract as a basis. *Level 3* reflects that the Mediator was not able to make the contractual parties agree on an amended contract. The Arbitrator collects all available evidence in order to reach the fairest decision possible. In case the Arbitrator's decision is accepted by all parties, the contract execution returns to *Level 1* and resumes with the arbitrated contract as basis. *Level 4* reflects the fact that the arbitration decision is not accepted by some of the contractual parties. The Mediator attempts to sanction the parties it sees as non-compliant, by invoking the corresponding Enforcer. *Level 5* reflects the fact that the DEM was not able to sanction the non-compliant parties. Legal procedures, outside of the realm of the e-contract management system, may be initiated.

After transaction completion, at the first three levels, the contractual parties are invited to provide feedback about each others performance. The feedback is collected by the Feedback Collection Centre (FCC) and is used to derive a reputation rating about each party in the system. Once the fourth level is reached, feedback is not collected from the contractual parties because it is assumed that the hostility between them will make the feedback highly biased and unreliable.

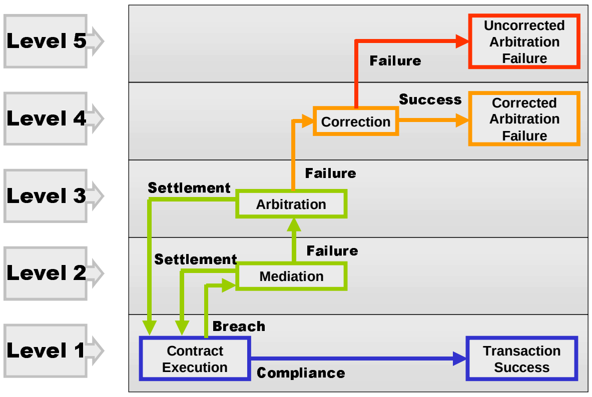

Figure 7: Contract enforcement process seen as a state diagram organised in layers of criticality

**4.3.3 Enforcer in virtual CCTs.** The functionality of the Enforcer for a virtual CCT is collectively realised by the enforcement capabilities of the participating CCT locals, coordinated by the Moderator of the virtual CCT. Since virtual CCTs span across organisational units and local managers act as intermediaries in any **me2ma** communication between local members and remote manager, any enforcement action takes place at a local level, although it may be initiated upon request of the remote Manager. In effect, this distribution of responsibility across CCT locals, reflects the fact that although contract enforcement in a virtual CCT is managed by its Moderator (which may be a TTP outside the virtual CCT or a functionality provided by entity undertaking the management of the virtual CCT), who is also initiating enforcement actions, the actual enactment is performed by the enforcer of each CCT local. That is, any enforcement options decided by the remote management need to be endorsed by the local management before they are performed on a member. This arrangement reflects the fact that it is the physical organisational units (i.e. CCT locals) who are contributing resources to a virtual CCT that effectively maintain high-level control of the contributed resources, even if they offer their management to their virtual CCT.

Although we do not mandate any particular solution to describe how an arbitrator arrives at this opinion, we believe that often this may well be based on second-hand evidence, and if some quantitative method can be used to guide this process, we propose the use of Subjective Logic [4]. The latter allows a peer to be able to encode and communicate its confidence in the validity of a non-performance statement or in the dependability of another peer's recommendation. This can be expressed in Subjective Logic notation, by means of an opinion $\omega_e^{LMonitor} = (b,d,u)$ encoding the peer's belief, disbelief and uncertainty. (See [6] for more on the use of Subjective

Logic for the evidence-based decision making underpinning discretionary contract enforcement.)

As explained in section 2.2, local managers act as intermediaries between local members and remote manager in **me2ma** communication within a virtual CCT. Hence, the opinion of the local members is passed to the remote manager as a recommendation via the local manager.

**4.3.4 Evidence based reasoning for Mediation and Arbitration within virtual CCTs.** Subjective Logic [4] provides a suitable mathematical foundation for trust-based decision making and combining first- and second-hand evidence in the presence of uncertainty. Confidence and trust modelling with subjective logic is useful when decision making is required. During monitoring no decision really has to be made, so using subjective logic does not help. It is during performance assessment, mediation and enforcement that when decisions have to be made, and that's when subjective logic can be useful. However, to correctly incorporate the reported evidence into a decision making mechanism, Moderator needs a scheme that reflects the monitoring configurations of the participating CCT locals.

In the following subsections we will explain how evidence provided from virtual CCT members or Monitor contributes to the decision making of the Moderator. We will use the concepts of an *opinion* and mainly two operators: *consensus* ⊕ and *discounting* ⊗.

In Subjective Logic notation we express the association of evidence to events by means of an opinion $\omega_e^B = (b, d, u)$ encoding the belief, disbelief and uncertainty of a peer B in the actual occurrence of some event e. The application of the consensus operator $\omega_e^B \oplus \omega_e^C$ effectively reduces uncertainty by collating evidence. This is particularly useful for amalgamating opinions based on partial evidence independently derived from data collected from different sensors. While $\omega_B^A \otimes \omega_e^B$ weights the evidence about an event *e* provided by an entity *B* against the belief element of $\omega_B^A$ representing belief in the competence of *B* to provide strong evidence about e. Disbelief and uncertainty in $\omega_B^A$ contribute to increasing the uncertainty of $\omega_B^A \otimes \omega_e^B$.

See also [4] and [5] for more on Subjective Logic and [6] for more on its application in the mediation and arbitration phases of our contract enforcement process.

**Incorporating evidence from Monitor:** When applying subjective logic it is normally required to provide first hand evidence as input, or else there might be hidden dependencies between the input parameters that could affect the correctness of the result. The Moderator should therefore receive the events and their confidence parameters as reported by first hand observers as far as possible. For this purpose we propose the following scheme for incorporating evidence provided by the different monitor configurations presented in subsection 4.2.1:

$$\omega_e^{DEM\,(VMonitor)} = \omega_e^{DEM\,(LMonitor_1)} \oplus .... \oplus \omega_e^{DEM\,(LMonitor_N)}$$

Where:

- *1,…, N* is an enumeration of the CCT locals which are contributing members to a virtual CCT;
- $\omega_e^{DEM\,(LMonitor_1)}, ...., \omega_e^{DEM\,(LMonitor_N)}$, respectively denote the Moderator's opinion derived on the basis of the (partial) evidence that has been provided by the monitoring capability of each CCT local *1,…, N.*

In the following paragraphs we examine how the above scheme is instantiated by different monitoring configurations of CCT locals. We will write $\omega_{LMonitor_1}^{DEM}, ..., \omega_{LMonitor_N}^{DEM}$ to denote the opinions respectively encoding the virtual CCT's Moderator confidence in the competence of monitoring in each CCT local[4].

Instantiation by a centralised monitor configuration is straightforward: if $\omega_e^{LMonitor_i}$ denotes the opinion of the centralised monitor of $i^{th}$ CCT local, then

$$\omega_e^{DEM\,(LMonitor_i)} = \omega_{LMonitor_i}^{DEM}\left[\omega_e^{LMonitor_i}\right] = \omega_{LMonitor_i}^{DEM} \otimes \omega_e^{LMonitor_i}$$

Note that $\omega_e^{LMonitor_i}$ is atomic and based on first-hand evidence.

Instantiation by a locally coordinated monitor is a little bit more complicated: if $\omega_e^{LMonitor_i^1}, ..., \omega_e^{LMonitor_i^k}$ respectively denote the opinions of the contributing coordinated monitor components (each of which is based directly on first-hand evidence), $\omega_{LMonitor_i^1}^{LMC_i}, ..., \omega_{LMonitor_i^k}^{LMC_i}$ respectively denote the opinions encoding the confidence of the local monitor coordinator in the competence of each monitor component, and $\omega_{LMC_i}^{DEM}$ denotes the confidence of the Moderator of the virtual CCT to the local monitor reporting consistently all evidence provided by the relevant sensors it orchestrates in the CCT local and its own trust metric for discounting the evidence collected by each sensor, then:

$$\omega_e^{DEM\,(LMonitor_i)} = \omega_{LMonitor_i}^{DEM}\left[\omega_e^{LMonitor_i^1}, ..., \omega_e^{LMonitor_i^k}\right] =$$
$$\omega_{LMC_i}^{DEM} \otimes \left(\left(\omega_{LMonitor_i^1}^{LMC_i} \otimes \omega_e^{LMonitor_i^1}\right) \oplus ... \oplus \left(\omega_{LMonitor_i^k}^{LMC_i} \otimes \omega_e^{LMonitor_i^k}\right)\right)$$

The above formula encodes the following process of incorporating evidence:

[4] As explained in [2], an agent is aware of its degree of trust in itself. Self-assessment underlies an agent's ability to seek external advice and to delegate or offer a task to another agent, so as to improve efficiency or reduce risk. In our case, although the Moderator may originate from one of the CCT locals which are contributing evidence, it may nevertheless maintain an opinion measuring the competence of its own local for Monitoring in a virtual CCT context.

1. for each local sensor, the trust metric provided by the local monitor coordinator is used to discount the evidence provided by that sensor;
2. the results of the above step 1, recording a weighted collection of independently generated evidence, are amalgamated into an overall opinion using the consensus operator;
3. the opinion encoding the confidence of the Moderator in the competence of the local monitor coordinator is used to discount the overall opinion derived in the above step 2.

Instantiation by a devolved monitor reduces to the previous case by making convention that, in the absence of any discounting to the opinion of a local sensor, forwarding the evidence collected by that sensor to the Moderator equals to showing absolute trust to that sensor. That is, absence of a local coordinator is the same as assuming a trivial imaginary local coordinator that shows absolute trust in the monitoring capability of each local member. Thus:

$$\omega_e^{DEM\,(LMonitor_i)} = \omega_{LMonitor_i}^{DEM}\left\lfloor \omega_e^{LMonitor_i^1},\dots,\omega_e^{LMonitor_i^k}\right\rfloor = \omega_{LMC_i}^{DEM}\otimes\left(\omega_e^{LMonitor_i^1}\oplus\dots\oplus\omega_e^{LMonitor_i^k}\right)$$

Where $\omega_{LMC_i}^{DEM}$ is the opinion presenting the confidence of the Moderator in the competence of the $i^{th}$ CCT local performing devolved monitoring.

**Incorporating opinions from CCT members:** If dispute arises or, during normal operation, for the purpose of assessing the performance of the CCT, the CCT moderator may seek the opinion of one or more CCT members about the completion of a task or any other statement related to the contract compliance.

Each CCT member may contribute its opinion encoded in a message, which is a (potentially distinguished) part of its **me2ma** communication and can be encoded in a special **me2ma** certificate. The relationship between the opinion contributed by each local member and the opinion of the Moderator associated with the remote management can be expressed in Subjective Logic notation by using a combination of the *consensus* $\oplus$ and the *discounting* $\otimes$ operators. (We assume CCT members to be able to form independent opinions based on first-hand evidence).

Since the corresponding CCT local manager acts as an intermediary between the local member and the remote manager in all **me2ma** communication, the opinion contributed to the CCT Moderator from a CCT member via its local manager is discounted by the CCT Moderator with its own opinion about the trustworthiness of the local manager. In addition, the local manager may choose to incorporate its own trust metric referring to its opinion about the competence of its local member. This ability of the local manager is particularly helpful since the local manager has a better overview of its local member's interactions in a variety of context than any of their remote managers. The local manager's opinion can be added to the **me2ma** message sent from the local member to the remote manager via the local manager. If the local manager does not provide an opinion, then absolute confidence of the local manger in the member is assumed. The CCT Moderator takes into account these opinions using the following scheme.

Assume that *1,…,N* is an enumeration of the CCT locals which are contributing members to a virtual CCT and the Moderator solicits opinions from members $me_i^1,\dots\dots,me_i^k$ CCT local. The Moderator's opinion based on the recommendation of $me_i^1,\dots\dots,me_i^k$ about a proposition α is expressed in Subjective Logic notation by the following formula:

$$\omega_{LM_i}^{DEM}\otimes\left(\left(\omega_{me_i^1}^{LM_i}\otimes\omega_\alpha^{me_i^1}\right)\oplus\dots\dots\oplus\left(\omega_{me_i^k}^{LM_i}\otimes\omega_\varepsilon^{me_i^k}\right)\right)$$

Note that the operator $\otimes$ does *not* distribute over $\oplus$. Consequently, the above formula is semantically and numerically different than:

$$\left(\omega_{LM_i}^{DEM}\otimes\omega_{me_i^1}^{LM_i}\otimes\omega_\alpha^{me_i^1}\right)\oplus\dots\oplus\left(\omega_{LM_i}^{DEM}\otimes\omega_{me_i^k}^{LM_i}\otimes\omega_\varepsilon^{me_i^k}\right)$$

In fact, the latter is wrong because it counts the same opinion $\omega_{LM_i}^{DEM}$ many times in a consensus, therefore violating the independence of evidence assumption for $\oplus$. Intuitively, should the latter formula be allowed the $i^{th}$ CCT local would be unfairly influencing the opinion of the Moderator. Consequently, instead of taking into account the opinions as solicited by the CCT members (originators), the Mediator needs to group together all opinions communicated via the CCT local manager (common intermediary).

## 4.4 Example

In the example, we look at the distributed CCT that represents a Collaborative Project. Project responsibilities are divided between the participant organisations, namely: a University, a Research Institute and a Company. Each of the institutions has several resources (people and/or services) directly participating in the project environment that is understood as a virtual CCT. It has been agreed that a Company will be in charge of the overall project. Therefore, its administration service (**Com**) acts as a **remote Manager** of the CCT, while the administration services of University and Research Institute (**Uni** and **RI**, respectively) act as an "**intermediate**" service, maintaining communication of its local clients with CCT manager. However, members of the Collaborative project are collaborating through P2P communication between them, without involvement of their local administrators. The following entities are directly involved in the project: computational resources at the University (*uni1*), simulation tool (*ri1*) and a researcher (*ri2*) at the Research Institute, and two researchers at the Company (*com1* and *com2*), Figure 8.

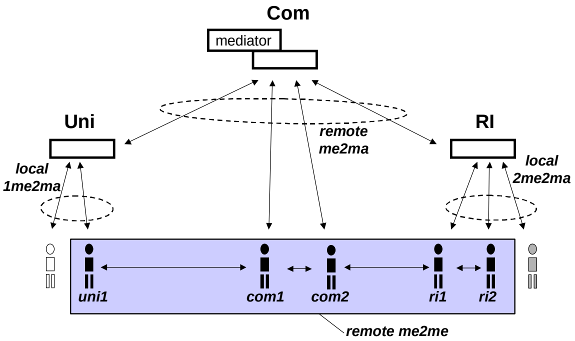

Figure 8: Virtual Group of Collaboration Project

| | | |
|---|---|---|
| *uni1* | Opinion: | $\omega^{uni1}_{non-a} = (0.7,0.2,0.1)$ |
| | Justification: | Following a recent upgrade, unavailability of resources at *un1* has been limited. |
| *Com1* | Opinion: | $\omega^{com1}_{non-a} = (0.1,0.7,0.2)$ |
| | Justification: | I have been using uni1 resources directly, without major problems |
| *Com2* | Opinion: | $\omega^{com2}_{non-a} = (0.2,0.2,0.6)$ |
| | Justification: | I have not been using the un1 resources regularly during this period. |
| *Ri1* | Opinion: | $\omega^{ri1}_{non-a} = (0.8,0.1,0.1)$ |
| | Justification: | During this period there was limited availability (40%) of the computational resources *uni1* |
| *Ri2* | Opinion: | $\omega^{ri2}_{non-a} = (0.6,0.1,0.3)$ |
| | Justification: | Sometimes resources at uni1 were unavailable, but ri2 did not use them frequently enough to be sure. |

Table 1: Member's opinions

| **Trustor:** **Trustee:** | Company Admin (COM) |
|---|---|
| com1 | $\omega^{COM}_{com1} = (0.6,0.2,0.2)$ |
| com2 | $\omega^{COM}_{com2} = (0.6,0.2,0.2)$ |
| uni1 | University Admin (UNI) |
| | $\omega^{UNI}_{uni1} = (0.6,0.2,0.2)$ |
| | Research Institute Admin (RI) |
| ri1 | $\omega^{RI}_{ri1} = (0.6,0.2,0.2)$ |
| ri2 | $\omega^{RI}_{ri2} = (0.6,0.2,0.2)$ |

Table 2: Inter-organisational trust values

| **Trustor:** **Trustee:** | Company Admin (COM) |
|---|---|
| UNI | $\omega^{COM}_{UNI} = (0.5,0.2,0.3)$ |
| RI | $\omega^{COM}_{RI} = (0.4,0.3,0.3)$ |

Table 3: Local trust values

The agreed service level conditions include that:

1) *results from the simulation tools must not be delayed more than 2 hours in total during one week*, and that
2) *unavailability of the computational resources will not be more than 4 hours in any 24h period.*

Assume that researcher *com2* estimates that results from the simulation tool were continuously delayed for over 3 hours/week over the period of last three week, and complains in order to preserve a good progress of the project. In response, simulation tool *ri1* claims that non-performance was due to frequent unavailability of the computational resources *uni1*.

The Moderator associated with the management of the Collaborative Project is therefore called upon and initiates a mediation process. The moderator asks each member of the CCT to provide their opinions about the unavailability of resources *uni1* (claim: *non-a*). Their opinions are summarised in Table 1.

The Moderator takes the opinions of the local members *com1* and *com2* (at the company) directly from the members and incorporates its own opinion about their competence in providing recommendations for this matter, represented respectively as $\omega^{COM}_{com1}, \omega^{COM}_{com2}$. See also Table 2.

Opinions from the other CCT members are received via their local managers, who act as their intermediaries in **me2ma** communication. Each of the local managers is thus acting as an advisor, using its own confidence in the competence of its local member to provide evidence for this matter in order to discount the actual evidence provided by each of its local members participating in the Collaborative project. Furthermore, the Moderator takes into account its trust in the management of the corresponding organization when incorporating all recommendations. Opinions capturing the above are presented in Table 2 and Table 3.

Overall, Mediator's opinion about resources unavailability is given by the following formula presented in Subjective Logic notation:

$$E\left(\omega^{COM:(com1,com2,UNI:(uni1),RI:(ri1,ri2))}_{non-a}\right) =$$

$$= \omega^{COMloc}_{non-a} \oplus \left(\omega^{COM}_{UNI} \otimes \omega^{UNIloc}_{non-a}\right) \oplus \left(\omega^{COM}_{RI} \otimes \omega^{RIloc}_{non-a}\right)$$

Where:

1. $\omega^{COMloc}_{non-a} = \left(\omega^{COM}_{com1} \otimes \omega^{com1}_{non-a}\right) \oplus \left(\omega^{COM}_{com2} \otimes \omega^{com2}_{non-a}\right)$ presents the Moderator's opinion about resources unavailability, based on the opinions of its local members;
2. $\omega^{RIloc}_{non-a} = \left(\omega^{RI}_{ri1} \otimes \omega^{ri1}_{non-a}\right) \oplus \left(\omega^{RI}_{ri2} \otimes \omega^{ri2}_{non-a}\right)$, presents the evidence about resources unavailability, provided by the local members of the research institute participating in the Collaborative Project;

3. $\omega_{non-a}^{UNIloc} = \omega_{uni1}^{UNI} \otimes \omega_{non-a}^{uni1}$ presents the evidence about resource provided by the local member of the university.

Note that even if members of a participating local (ie. Company, University, Research Institute) each sends opinions to the Moderator associated with the remote administration, this Moderator is grouping all recommendations originating from the same local. As explained in section 4.3.4, this is done in order to ensure that evidence communicated via the same intermediary is grouped together and then discounted with the trust in that intermediary before taken into consideration.

The overall Moderator's opinion about non-availability of the resources during the critical period is: $\omega_{non-a}^{COM} = (0.748{,}0.229.0.021)$. Probability expectation [4] $E\left(\omega_{non-a}^{COM:(com1,com2,UNI:(uni1),RI:(ri1,ri2))}\right) = 0.760$ presents the result of Moderator's judgement[6].

Assuming that a minimum value of 0.75 is defined as a threshold for the mediator to pronounce a decision, the conclusion is that a contract breach occurred due to computational resources non-availability, and not due to delayed delivery of simulation tool results.

Unfortunately the Collaborative Project contract does not provide a specific sanctioning and recovery mechanism for this case. Consequently, the Mediator and Negotiator capabilities of the Company's management are called in and an recovery (together with a contract amendment are negotiated.) The University agrees to pay a fine of £1,000 and initiate a further update in order to improve availability of its resources. The Moderator of company's management delegates to the enforcer of the University to bring about the software update.

## 5 Conclusion & Further Work

Security, trust and compliance to the collaborative business agreements are the main prerequisites for successful functioning and operation of Virtual Organizations (VO).

In this paper we explored the problem of supporting secure and dynamically evolving collaborations between parties involved in a VO throughout its life-cycle. We proposed a new paradigm based on the integration of the *Closed Collaboration Teams* (**CCT**) paradigm and *Business Contract Architecture* (**BCA**), which we expect to provide the basis for an ecosystem supporting the instantaneous creation and dynamic evolution of secure collaborations across enterprise boundaries and in compliance with electronic contracts whose enactment is autonomic.

Some of the functionalities described in the paper are being tested in a simulation model built using *OPNET Modeler*[5]. In particular, the basic group collaboration architecture underpinning the CCT concept, and basic security mechanisms, comprising: 1) distribution and enforcement of security policies, and 2) monitoring and profiling of user behaviour through anomaly detection, are currently being implemented. By further extending our simulation model we also plan to test some of the enhanced BCA functionality discussed in this paper. Longer-term plans include the development of an operational implementation as in the context of a larger Enterprise Grids infrastructure.

## Acknowledgements

The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Industry, Science & Resources); the UK Council for the Central Laboratory of the Research Councils (CCLRC); Information Society Technologies grants IST-2001-35464 (Grid-based Application Service Provision) and IST-2001-34910 (iTrust: Working Group on Trust Management in Dynamic Open Systems) of the Framework Programme 5 of the European Commission.

## References

[1] T. Dimitrakos, I. Djordjevic, B. Matthews, J. Bicarregui, C.I. Phillips - Policy Driven Access Control over Distributed Firewall Architecture. Proc. Policy 2002. IEEE CS 2002.

[2] T. Dimitrakos - System Models, e-Risk and e-Trust. Proc. IFIP I3E-1, Kluwer Academic Publishers 2001.

[3] I. Djordjevic, C. Phillips - Certificate-Based Distributed Firewalls for Secure E-Commerce Transactions. FITCE Congress, 2001; J. of IBTE, vol.2, part3, pp. 14-19.

[4] I.A. Jøsang - A Logic for Uncertain Probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 9(3):279–311, June 2001.

[5] A. Jøsang - The Consensus Operator for Combining Beliefs. Artificial Intelligence Journal, 142(1-2);157-170, Oct.2002.

[6] Z. Milosevic, A. Josang, T. Dimitrakos, M.A. Patton – Discretionary Enforcement of Electronic Contracts. Proc. EDOC '02. pp(s): 39 -50. IEEE CS 2002.

[7] Z. Milosevic - Enterprise Aspects of Open Distributed Systems. PhD thesis, Computer Science Dept. The University of Queensland, October 1995.

[8] Z. Milosevic, D. Arnold, L. O'Connor - Inter-enterprise contract architecture for open distributed systems: Security requirements. Proc. of WET ICE'96 Workshop on Enterprise Security, Standford, June 1996

[5] OPNET and OPNET Modeler are registered trademarks of OPNET Technologies, Inc.