GOLD Infrastructure for Virtual Organisations

Periorellis P., Cook N., Hiden H., Conlin A., Hamilton M.D.,
Wu J., Bryans J., Gong X., Zhu F., Wright A.

Panayiotis.Periorellis@ncl.ac.uk

May 2006

North East Regional e-Science Center,
School of Computing Science,
Claremont Tower,
University of Newcastle Upon Tyne,
Newcastle, NE1 7RU, UK

Abstract

The principal aim of the GOLD project (Grid-based Information Models to Support the Rapid Innovation of New High Value-Added Chemicals) is to carry out research into enabling technology to support the formation, operation and termination of Virtual Organisations (VOs). This technology has been implemented in the form of a set of middleware components, which address issues such as trust, security, contract monitoring and enforcement, information management and coordination. The paper discusses these elements, presents the services required to implement them and analyzes the architecture and services.The paper follows a top down approach starting with a breif outline on the architectural elements, derived during the requirements engineering phase and demonstrates how these archtiectural elements were mapped onto actual services that were implemented according to SOA principles and related technologies.

1 Introduction to GOLD

GOLD (an EPRSC Pilot project) aims to assist virtual organisations to form, operate and terminate by privinding a set of architectrural components that are dependable while at the same time flexible regarding their adaptation and usage. The project itself went through a thorough process of requirements engineering by investigating the actual needs of potential virtual organisations. It has therefore delivered an architecture that addresses those needs that can be broadly categorised in terms of security, workflow, storage and contract management requirements. The purpose of this paper is to discuss the architectural components while at the same time provide some detail regarding the actual implentation. The paper is structured as follows. Section 2 provides a brief outeline of the architectural elements that comprise the GOLD infrastructure. Section 3 which forms the core of this paper describes the services that were imlpemented to reflect the archtiectural elements discussed in section 2. The paper concludes with section 4. It should be noted here that the majority of requirements were gathered by researching the potential of virtual organisations, forming within the chemical development sector. The team was in close contact with a number of companies from that domain which helped capture the more intricate details of the infrastructure. Nonetheless the infrastructure is not tailored to that specific domain, as it is build as a generic set of services that can be adapted in a flexible manner.

2 Architecture

The GOLD Middleware architecture has primarily been derived through the application of Soft Systems Modelling in addition to a number of interviews that were conducted with companies wihtin the chemical development sector [Periorellis et. al. 2006]. Some of the early findings suggested that the infrastructure needs to be flexible, adaptable and capable of coping with the dynamic characteristics of VOs. In addition, it is undesirable to impose unnecessary constraints on the potential for VO formation by dictating the specifics of the various supporting technologies the entities are required to deploy in order to participate in a VO. Having used these models as a guide we have derived the following architectural that comprise GOLD [Hiden et. al. 2005]. Figure 1 below shows the main elements identified following the analysis of the

SSM model.

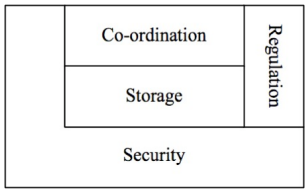

Figure 1 - Architectural elements

To support such a dynamic approach, including the need for late binding and loose coupling to actual implementation technologies, a Service Oriented Architecture (SOA) [Erl 2004] based upon standard Web Services [Skonnard 2002] has been identified as the most suitable means of implementing the GOLD Middleware. Web Services make it possible to use a variety of standards and protocols, and allow the integration of different software applications running on heterogeneous systems. It is important to note, however, that the architectural elements described in the following sections do not necessarily map directly to individual services; rather, they represent high level areas of functionality that require one or more physical services to support them. The security element is paramount encompasses mechanisms for information exchange, access to ressources, user authentication and autorisation. The quantity of information generated in a virtual organisations is significant. This information needs to be stored such that it is available to, and searchable by, correctly authenticated and authorised VO participants. Central to the storage aspect of the GOLD Middleware is the information model describing the topology of the VO and the data and document types that can be exchanged between participating entities. This is a key aspect of the system as it supports the extensibility needed to allow the infrastructure to be tailored to different problem domains. The coordinaation element emphasises the need for planning within a VO. Tasks are coordinated, and will either be performed manually or automatically. Therefore Middleware platforms need to support not only the enactment of pre-determined workflows, but also provide a flexible environment that does not follow a fixed workflow. The Regulation aspect of the architecture aims to ensure that entities who interact within a VO are able to exercise their rights and that, at the same time, they meet their obligations to one another and to any relevant regulatory body.

3 Service Implementations

To support the architectural elements introduced in Section 2, the GOLD project has derived and implemented a number of core services. Figure 2 shows these GOLD services and their relationship to the architectural elements.

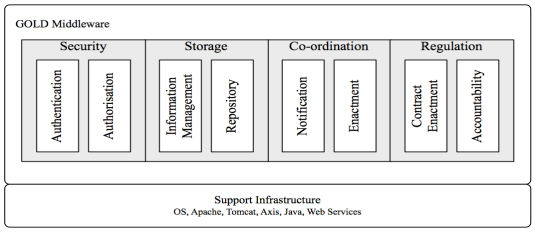

Figure 2 - Web Service oriented infrastructure

This section of the paper describes the technical implementation details of the architecture and describes the set of services currently implemented.

3.1 Security

When granting personnel from an external company access to internal resources it may be necessary to restrict access to those resources. To address this issue, the security element of the GOLD architecture is implemented in the form of authentication and authorisation services which enable members of a VO to define the roles relevant to their project in a central or federated manner. These roles can then be assigned access rights to resources locally, based on the work required to be carried out. In addition these rights can be updated depending upon the stage of a project allowing fine-grained access control.

3.1.1 Authentication

Authentication describes the process of securely establishing and verifying identities of network subjects which may take the form of users, agents, registered services or components. The objectives of the authentication mechanism of the GOLD Middleware are to make sure that only the correct participants enter and operate within the VO and to allow the participants to interact freely (within the range set by the access control policies) with the various services and resources provided by the VO. During the lifetime of a VO its participants will be required to share resources and hence access to those resources will require crossing of organisational boundaries. Clearly, expecting a VO participant to log in several times in order to carry out a task that is part of the same activity is not productive. Several approaches have been proposed, most notably Microsoft Passport [MS Passport 2005] and the Liberty Alliance Group [Liberty 2005]. The GOLD infrastructure supports privacy of a user's own information as long as there is a traceable link between the federated identity and their credentials. For reasons such as data protection and privacy the infrastructure issues a federated identity valid only within the VO. Participants can therefore retain their privacy as the federated identity does not identify the real identity of the participant. This implies that the infrastructure maintains a traceable link between the

federated identity and the real identity of the participant allowing both accountability and privacy to be supported. Secure message exchange between the VO participants is achieved by exchanging security tokens carried within the headers of the exchanged messages. The federation service signs and attaches these tokens to SOAP message headers according to the WS-Security specification [WS-Security 2004]. The specification supports various token structures such as X.509 certificates, Username or XML-based tokens which have been customised, in the GOLD Middleware, to support SAML [OASIS 2004] assertions. The structure of a message carrying a token and the lifecycle of such a token from request to validation is shown in Figure 3.

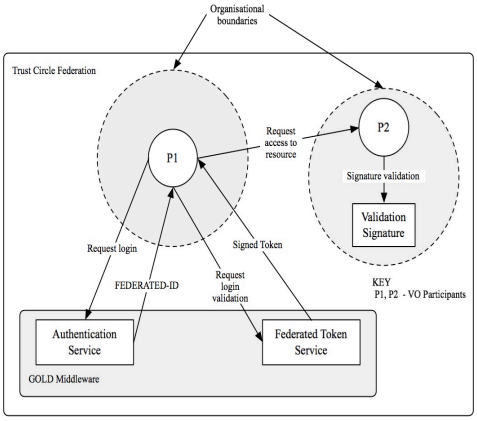

Figure 3 - Security token lifecycle

In the body of the message carries the actual message while the header is structured according to WS-Security specification and carries a security token formatted in one of the standardised styles described earlier. The above description implies that a federation has been established and that there is a direct trust relationship between the federation and the participants. If this direct relationship is not present the above model fails. To alleviate this problem the GOLD Middleware makes use of the WS-Trust [2005] specification that offers a mechanism for managing brokered trust relationships.

3.1.2 Authorisation

In earlier publications [Periorellis et al. 2004] we discussed the advantages and disadvantages of access control models, ranging from active control lists to role and task based systems. We concluded that the dynamic nature of virtual organisations makes it necessary for any VO infrastructure to support a mechanism that deals with dynamic rights activation and de-activation. In dynamic systems such VOs, both the topology of the VO (parties that form it and links between them) and its workflow are subject to change. Therefore static rights assignment prior to a workflow does not capture the eventuality of a party leaving, added or any alterations to the workflow itself. Several authors have elaborated on this issue [Coulouris, G., et al. 1994, Roshan, K., T., et al. 1997, Sandu, R., S., et al. 1996, Thomas R., K., 1997]. In addition, given the sensitivity of the information that may be shared in some VOs, (which raises concerns regarding competitive advantage) parties are not expected to be assigned a single set of rights that would last throughout the duration of a VO. It is more likely that VO participants would agree limited or gradual access to their resources depending on workflow progress. In GOLD we want to be able to restrict role access on resources depending on the execution context. In a virtual organization the execution context can be regarded as the particular project or the goal that all participants have come together to achieve. In a virtual organization there can be many projects with various participants resulting to complicated interrelationships between them. For example, some of them may play the same role in various projects, carrying out the exact same tasks, or have different roles within the same project depending on the performed task. The question we raise is 'Should a role have the same permission and access rights throughout the set of similar or even identical projects and the tasks within those projects?' Our view is that in dynamic access control systems we should separate roles from role instances. To support this, we present the role instances as parameterised roles. Those parameters can be used to express relations to distinguish the role with task-specific or project-specific privileges from the general role. Also some roles may be relational, e.g. originator(document, project) and it may be necessary to enforce separation of duties for such purposes as ensuring that the originator of a document cannot also sign it off. Policy on sign-off could include the possession of certain qualifications in combination with an organisational position, for example the manager can sign off a document provided he/she also has a chemist qualification and is not the originator of that document. Different role instances may require different permissions and indeed additional levels of authorization depending on the project and task in which they are active. To be able to handle such cases, GOLD needs to support adequately fine-grained access control mechanisms. Parameterisation of roles supports both relational roles and fine-grained access control In order to raise the levels of trust in those cases, one needs to make sure that adequately fine grained access control mechanisms are supported. Granularity refers to the level of detail for which one can define access rights. Fine grained permissions are needed for instances of roles as well as instances of objects. For example, a chemist role may be granted access to chemical documents but we do not however wish to grant access to all chemical documents produced by the system. Instead, we want any access permissions granted to the chemist role to be project-specific (e.g., the instance of a particular collaboration) as well as task-specific (e.g., the instance of a particular pre-defined set of activities). So the management of roles and access permissions in

GOLD needs to be integrated with the management and monitoring of dynamic service level agreements or contracts between the participating services. The contracts can capture the expectations from specific tasks, using pre- and post-conditions. Permissions for roles can be activated and de-activated based on the progress of the monitored contracts.. Rights should not be automatically assumed upon role assignment. Instead they should be granted gradually, as the workflow progresses, prohibiting access to parties that may be part of a workflow but are not currently enacting the task for which the access right is relevant. Equally important is that rights may become more restricted with workflow progress, thus the achievement of certain milestones may trigger a permanent change of access rights. Assume a certain project has four phases, Project Evaluation, Route Development, Process Development and Technology Transfer, with a milestone at the end of each phase. Each milestone is an event that will be notified by the Coordination service (see section 3.2 below), to which the Security service subscribes. On receiving this event, the Security service will make appropriate changes to the rights associated with role instances: for example, at the outset of the project four roles - Senior Manager, Financial Analyst, Project manager and Senior Chemist - all have Read and Write permissions on documents within the Marketing Dossier. On achievement of the Route Development milestone, Write permission on these documents is removed from the Financial Analyst and Senior Chemist roles. When the Process Development milestone is reached, write permission on these documents is also removed from the Project Manager role. Notification by the Coordination service to the Security service also supports the dynamic revocation of roles, prohibiting access to parties that are not fulfilling their obligations or who are no longer in the given organisational role. Given these requirements, there are several functionalities which the infrastructure has to support including:

- a common language for expressing authorisation policies that is understood by all participants;
- a protocol for expressing policies and rules that is understood by all participants;
- a protocol for transferring/communicating these policies between VO participants;
- a centralised policy repository;
- a verification component which ensures policy consistency.

Access control policies within the GOLD Middleware are expressed using XACML which is a Web Services standard [OASIS 2003]. XACML has an RBAC profile, which we are extending to provide for parameterised roles .

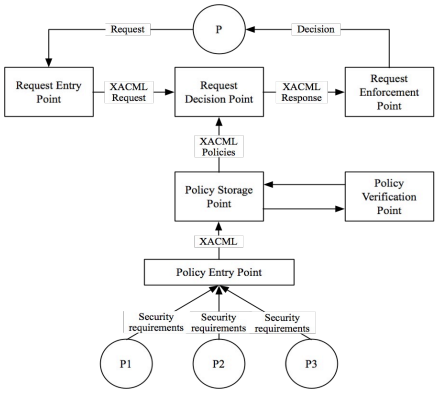

Figure 4 - Access control service architecture

In Figure 4 which illustrates the GOLD access control service, **P** represents a participant and the remainder of the boxes represent the services that have been implemented. VO participants (i.e. **P1**, **P2** and **P3**) express their security requirements using the service interface. This interface provides a user-friendly policy language and also supports consultation of the workflow model, to enable policies to be expressed at the level of specific tasks or subprocesses. These requirements form a set of policies that describe how individual participants need their resources to be protected. The policies for the same project or workflow are placed in temporary storage where a verification service, will validate them for logical inconsistencies. The verification service guarantees that no exceptions will be thrown during VO operations as a result of policy mismatches. Any possible mismatches can be highlighted and compromises negotiated between participants prior to the commencement of VO projects. Assuming that a participant, requests access to a resource, several services coordinate the process of expressing a request and providing a response. Given the wide range of policies that may be required to fully specify the access control requirements within a VO and the fact that there is no single authority governing these policies (the majority of which will stem from participants' requirements on how they want to protect their resources), verification is needed to ensure that there are no logical inconsistencies.

3.2 Coordination

A key aspect of collaborative working is to have some means of ensuring that all interested/involved VO particpants receive coordination messages and notifications. It is also necessary to ensure that tasks are performed at the appropriate time and with the appropriate participants. This leads to a requirement for Notification and Enactment services. VO participants are informed about certain events that take place within the VO through the Notification service. The GOLD infrastructure has adopted the simpler WS-Eventing model which specifies a simple

architecture for subscribing, producing and delivering notification messages. It supports typical subscription operations such as subscribe, unsubscribe and renew, while the delivery mechanism is based on the pull model similar to OMG's CORBA notification model [Bolton 2001]. When the participant subscribes to GOLD's notification service, the service includes information regarding the Subscription Management service in its response. Subsequent operations, such as getting the status of, renewing and unsubscribing, are all directed to the subscription manager. The source (producer) sends both notifications and a message signifying the end of registered subscriptions to the sink (consumer). To provide notification services the GOLD Middleware makes use of NaradaBrokering [Pallickara and Fox 2003] which is a mature, open source, distributed messaging infrastructure. NaradaBrokering can support a number of notification and messaging standards, notably: JMS [JMS 2001], WS-Eventing and WS-Notification. It is, therefore, suitable for intra- and inter-organisational notification. In the context of a VO, a significant advantage of a notification service built on NaradaBrokering is the flexibility of deployment options. The service could be deployed as a stand-alone service at a single site or, alternatively, as a peer-to-peer network of Narada brokers offering a federated notification service. For example, the notification service could be distributed across a set of Trusted Third Parties (TTPs) that support the VO or across the members of the VO itself. In either deployment NaradaBrokering provides scalable, secure and efficient routing of notifications. The Enactment service operates in conjunction with the core GOLD Middleware services to provide support for coordination and workflow. The main components of the service are the workflow editor and the workflow enactment engine, see Figure 5. In addition, there are custom workflow tasks for manipulating VOs and managing documents.

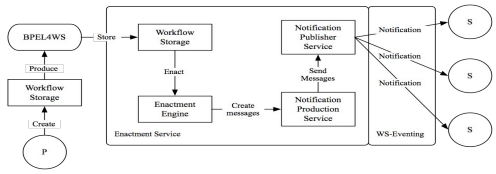

Figure 5 - Enactment service

In Figure 5, participant P describes a workflow using a workflow editor and saves the output in a storage facility. Workflows may be represented graphically using an editor that subsequently creates an XML output document structured according to the BPEL4WS (Business Process Enactment Language for Web Services) specification [BPEL4WS 2002]. The technology provides an XML schema (currently being considered as an OASIS standard) for describing tasks, roles, inputs, outputs, pre and post conditions associated with each task. During workflow enactment, the workflow engine retrieves BPEL4WS documents and executes them, subsequently sending any notifications required to subscribers (S) interested in the state of the coordination activities. GOLD adopted ActiveBPEL [Emmerich et al. 2005] for workflow execution. The Workflow Enactment service responds to requests by initiating workflows to co-ordinate interactions between VO members. For example, the process of approving a chemical safety study may involve passing this study to several approvers, each of which must sign the document before it is considered complete. The enactment engine notifies the participants about events and required activities, depending on the topics participants have registered for. For situations where it is not necessary to co-ordinate the detailed activities of VO participants, the GOLD Middleware can be used to provide an abstraction with a lower degree of integration, for instance a shared space that contains projects that are divided into discrete tasks. Each of these tasks can contain a set of data, which can be accessed and edited by participants with the correct security credentials. These documents can include any of the information types supported within the VO, including the results of any enacted workflows, and any workflows that are still on-going.

3.3 Regulation

Regulation helps govern interactions between parties, ensuring that participants' rights (in terms of the resources they are allowed to access) are properly granted and that obligations are properly dispatched (such as making resources available to others). This is achieved by the use of contracts and contract enforcement mechanisms as well as monitoring mechanisms for auditing and accountability.

3.3.1 Contract Enactment

Each member of a VO requires that their interests are protected, specifically:

- that other members comply with contracts governing the VO;
- that their own legitimate actions (such as delivery of work, commission of service) are recognised;
- that other members are accountable for their actions.

To support this, the GOLD Middleware records all activities to monitor for compliance with the regulatory regime. Furthermore, critical interactions between VO participants should be non-repudiable (no party should be able to deny their participation) and the auditing and monitoring functions must be fair (misbehaviour should not disadvantage well-behaved parties). For example, a business contract governing the scenario described in Section will specify sequencing constraints on the production of documents such as the requirements, recipe, thermal analysis and scale-up analysis. It will also require that the safety of the scaled-up process is assured, hence

the requirement that the the Scale-up company employ the Thermal Safety company to provide an analysis and validation of the potential exotherm that was identified during the scale-up studies. In a complex natural language contract of the form typically negotiated between business partners, there may in fact be ambiguities that, given the sequencing constraints, could lead to deadlock during the chemical process development project. There is, therefore, a need to produce an electronic version of the contract that can be model-checked for correctness properties (e.g. safety and liveness). Having developed a contract that is free of ambiguities, it should then be possible to derive an executable version to regulate an interaction at run-time. That is, the business messages exchanged between participants during the process development should be validated with respect to the executable contract to ensure that they comply with contractual constraints. To hold participants to account, and to be able to resolve subsequent disputes, these message exchanges should be audited. In the scenario described in Section , the Scale-up company sends the Supplier company the scale-up model. The delivery of this document should be validated against the contract to ensure any pre-requisite obligations have been fulfilled. To safeguard their interests, the Supplier company will require evidence that the model originated at the Scale-up company. Similarly, the Scale-up company will require evidence that they fulfilled their obligation to deliver the model to the Supplier company.

Given these requirements, we identify two main aspects to contract enactment and accountability:

- high level mechanisms to encode business contracts so that they are amenable to integrity and consistency checking and in order to derive an executable form of contract;
- a middleware infrastructure that can monitor a given interaction for conformance with the executable contract - ensuring accountability and acknowledgement.

To address the first aspect, Section 3.3.1.1 provides a summary of work on the derivation of electronic contracts and deployment models for contract mediated interaction. This work appears in Molina et al. [2005], which also presents related work. The GOLD project extends this work to enact business contracts using infrastructure for accountability and non-repudiation as an enforcement mechanism. Section 3.3.2 presents this infrastructure and shows how it addresses the second aspect identified above. This paper is concerned with monitoring and enforcement of business operation clauses, of equal importance is the monitoring of the levels of Quality of Service (QoS) offered within a VO. This concerns the collection of statistical metrics about the performance of a service to evaluate whether a provider complies with the QoS that the consumer expects. Molina et al. [2004] examine this aspect of regulation and related work.

3.3.1.1 Contract-mediated interaction

The rules in a conventional, paper-based contract express what operations business partners are:

- permitted to perform if deemed necessary to fulfill the contract;
- obliged to perform to ensure contract fulfillment;
- prohibited from performing as these actions would be illegal, unfair or detrimental to the other partners.

In addition, rules may stipulate when and in what order the operations are to be executed. To form and have automatic management of partnerships within a VO, electronic representations of contracts must be used to mediate the rights and obligations that each member promises to honour. In the worst case, violations of agreed interactions are detected and all interested parties are notified. In order to support this, the original natural language contract that is in place to govern interactions between participants has to undergo a conversion process from its original format into an executable contract (x-contract) that works as a mediator of the business conversations. This conversion process involves the creation, with the help of a formal notation, of one or more computational models of the contract with different levels of details. To achieve these objectives, the Promela modeling language [Holzmann 1991] is used to represent all the basic parameters that typical business contracts comprise, such as permissions, obligations, prohibitions, actors (agents), time constraints, and message type checking. The Promela representation can be validated with the help of the accompanying Spin model-checker tool [Holzmann 2004]. For example, model-checking the Promela representation can improve the original natural language contract by removing various forms of inconsistency as discussed in Solaiman et al. [2003]. This implementation-neutral representation can be refined to include technical details such as acknowledgment and synchronisation messages. The details will vary depending on specific implementation techniques and standards that are adopted. This implementation specific representation can then be used for run-time monitoring. Conceptually, an x-contract is placed between VO members to regulate their business interactions. In terms of the interaction model, the x-contract may be reactive or proactive. A reactive x-contract intercepts business messages, validates the messages and rejects invalid messages. A proactive x-contract drives the cross-organisational business process by inviting VO members to send legal messages of the right type, in correct sequence, at the correct time etc. Deployment can be either centralised or distributed. This leads to four deployment models:

- *Reactive central* - where all messages are intercepted by a centralised x-contract (at a TTP, for example) that is responsible for forwarding just the legal messages to their intended destination.

- *Proactive central* - where a centralised x-contract coordinates the business process on behalf of VO members and triggers the exchange of legal messages.
- *Reactive distributed* - where the x-contract is split into separate components that can be used to validate just those messages sent to an individual VO member and to reject illegal messages sent to that member.
- *Proactive distributed* - a distributed version of proactive central that coordinates the legal participation of each member in the business process.

Distributed deployments face the difficult challenge of keeping contract state information synchronised at both ends. For example, a valid message forwarded by the buyer's x-contract could be dropped at the seller's end because intervening communication delays render the message untimely (and therefore invalid) at the seller side. State synchronisation is necessary to ensure that both the parties agree to treat the message as either valid or invalid. One approach that uses a non-repudiable state synchronisation protocol [Cook et al. 2002] is described in Molina et al. [2003]. The GOLD Middleware used to invoke validation with respect to a contract at runtime is discussed below in Section 3.3.2.

3.3.2 Accountability

This section focuses on accountability for the delivery of a single business message. However, this validated and non-repudiable message delivery can then be used as a building block for contract monitoring and enforcement of the kind envisaged in Section 3.3.1.

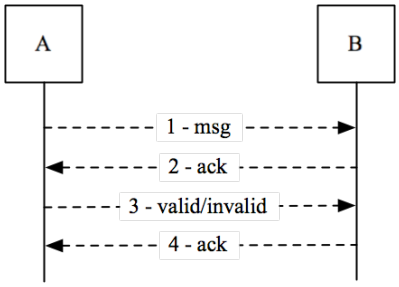

Figure 6 - Business message delivery with acknowledgements

Figure 6 shows the delivery of a business message and associated acknowledgements. Typically, for each business message, there should be an immediate acknowledgement of receipt indicating successful delivery of the message. Eventually, a second acknowledgement indicates whether the original business message is valid (or invalid) in the context of the given interaction. Finally, the validation message is acknowledged in return. Validation of the original business message (performed at **B**) can be arbitrarily complex. For example, it may simply involve verification that a message is syntactically valid and in the correct sequence with respect to a contract. Alternatively, a message may require validation with respect to more complex contractual conditions or with respect to local application state. Triggering validation at the level of business message delivery has the potential to allow specialisation of an application to meet the constraints of different regulatory regimes. Web Services are increasingly used to enable B2B interactions of this kind. However, there is currently no support to make the exchange of a set of business messages (and their associated acknowledgements) both fair and non-repudiable. A flexible framework for fair, non-repudiable message delivery has therefore been developed. The Web Services implementation of this framework comprises a set of services that are invoked at the middleware level and so enable the Web Services developer to concentrate on business functions. The GOLD Middleware renders the exchange of business messages fair and non-repudiable. Arbitrarily complex, application-level validation is supported through the registration of message validators. The framework is sufficiently flexible to adapt to different application requirements and, in particular, to execute different non-repudiation protocols to meet those requirements.

3.3.2.1 Basic concepts

Non-repudiation is the inability to deny an action or event. In the context of distributed systems, non-repudiation is applied to the sending and receiving of messages. For example, for the delivery of a message from A to B the following types of non-repudiation may be required:

- *NRO* - B may require Non-Repudiation of Origin of the message, i.e. irrefutable evidence that the message originated at A;
- *NRR* - A may require Non-Repudiation of Receipt of the message, i.e. irrefutable evidence that B received the message.

Non-repudiation is usually achieved using public key cryptography. If A signs a message with their private key, B can confirm the origin of the message by verifying the signature using A's public key, and vice versa. An additional requirement is that at the end of the interaction no well-behaved party is disadvantaged. For example, consider the selective receipt problem where a sender provides NRO but the recipient does not provide the corresponding NRR. This problem is addressed by the fair exchange of items where fairness is the property that all parties obtain their expected items or no party receives any useful information about the items to be exchanged [Markowitch et al. 2002]. Kremer et al. [2002] provide a survey of protocols to achieve fair, non-repudiable exchange of messages. The following discussion is based on the use of an in-line TTP to support the exchange. However, our execution framework is not restricted to this class of protocol.

3.3.2.2 Overview of approach

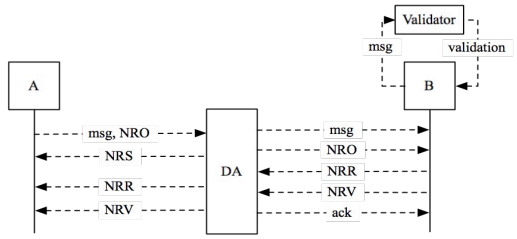

Figure 7 - Executing a business interaction through a delivery agent

Figure 7 introduces a Delivery Agent (DA), or inline TTP, to the interaction shown in Figure 6. Four types of evidence are generated:

- *NRO* - Non-Repudiation of origin that *msg* originated at A;
- *NRS* - Non-Repudiation of submission to the DA of *msg* and NRO;
- *NRR* - Non-Repudiation of receipt of *msg* by B;
- *NRV* - Non-Repudiation of validation, valid or otherwise, as determined by validation of *msg* by B.

A starts an exchange by sending a message, with proof of origin, to the DA. This is the equivalent of Message 1 in Figure 6 with the NRO appended. The DA exchanges *msg* and NRO for NRR with **B** (before application-level validation of *msg*). Then the DA provides NRR to **A** equivalent to Message 2 in Figure 6. Subsequently, **B** performs application-level validation of *msg* (as in Message 3 of Figure 6 and provides NRV to the DA. The DA, in turn, provides NRV to **A** and provides acknowledgement of NRV to **B**. The exact sequence of the exchange will be dictated by the actual protocol used and should not be inferred from Figure 7.

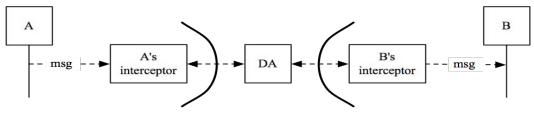

Figure 8 - Interceptor approach

As shown in Figure 8, our approach is to deploy interceptors that act on behalf of the end users in an interaction. An interceptor has two main functions:

- to protect the interests of the party on whose behalf it acts by executing appropriate protocols and accessing appropriate services, including TTP services;
- to abstract away the detail of the mechanisms used to render an interaction safe and reliable for its end user.

In this case, the mechanism used communicates through the DA. It is the responsibility of the DA to ensure fairness and liveness for well-behaved parties in interactions that the DA supports. The introduction of interceptors means, that as far as possible, **A** and **B** are free to concentrate on application level concerns while their interaction is rendered fair and non-repudiable.

3.4 Storage

The storage element addresses the need to store, manage and access information. In addition there is a requirement to be able to determine how a piece of information was derived. The Information Management and Repository services meet this need by providing configurable information storage and logging/auditing functionality.VOs must control and manage the exchange of information between the participants, and the role of the Information Management service in the GOLD Middleware is to support this exchange in three ways:

- to ensure a common structure and meaning for information shared across the VO;
- to provide information services and tools to support the controlled exchange of information according to the policies and contracts that are in place within the VO;
- to extract value from the information stored during the lifetime of a VO.

To support the information management requirements of VOs the GOLD Middleware provides an Information Model that defines the structure and meaning of information shared by its participants. This model can be divided into three categories:

- *Generic* - represents information that is required by all VOs. This includes descriptions of the VO structure, the participants, the tasks being performed, security policies etc. The services that make up the generic GOLD VO infrastructure (i.e. those comprising the security, coordination and regulation architectural elements) all exchange information defined in this category of the information model.
- *Domain specific* - within a particular domain, there are types of information that are generic across a broad range of VOs.
- *Application specific* - information in this category represents specialist information describing a particular domain.

This information model is based on the myGrid information model [Sharman et al. 2004], which was designed to support VOs in the bioinfomatics domain.

4 Conclusions

GOLD middleware offers a set of services that can be used to assist in the formation, operation and termination of virtual organisations. The aim of the project and the proposed architecture is to offer VO developers the flexibility to configure the VO

according to their requirements without imposing too many constraints or imposing what and how it should be done. In this limited space we touched on 4 fundamental architectural elements and discussed in turn how they could be implemented. Adhering to certain principles regarding privacy and trust we devised a security policy for authorisation and authentication that is based primarily on current WS standards. Virtual organisations bring together a number of independent entities with the aim to collaborate in achieving a common goal. This creates the need for some form of coordination regarding the message exchanges between those entities. Coordination therefore is a key aspect of collaborative working. Participants have to remain informed of certain events and It is also necessary to ensure their obligations are dispatched at the appropriate time. The paper showed how regulation helps govern interactions between parties, to ensure that obligations are properly dispatched and rights are properly granted by the use of contracts and contract enforcement mechanisms as well as monitoring mechanisms for auditing and accountability. Finally shed some light was on the issue of storage and information management and as such several broad requirements and implementation strategies were discussed.

References

BOLTON, F. 2001, Pure CORBA, SAMS, ISBN 0672318121

BPEL4WS. 2002. *BPEL4WS V1.1 specification*. ftp://www6.software.ibm.com/software/developer/library/ws-bpel1.pdf.

CHADWICK, D. and OTENKO, A. 2003. The PERMIS X.509 role based privilege management infrastructure. *Future Generation Computer Systems*, 19 , 2, 277-289.

CHECKLAND, P. B. and SCHOLES, J. 1990. *Soft Systems Methodology in Action*, John Wiley and Sons, Chichester.

CONLIN, A.K., ENGLISH, P.J., HIDEN, H.G., MORRIS, A.J, SMITH, R. and WRIGHT, A.R. 2005. A Computer Architecture to Support the Operation of Virtual Organisations for the Chemical Development Lifecycle. In Proceedings of European Symposium on Computer Aided Process Engineering, (ESCAPE 15), 1597-1602.

COOK, N., SHRIVASTAVA, S. and WHEATER, S. 2002. Distributed Object Middleware to Support Dependable Information Sharing between Organisations. In Proceedings of IEEE Int. Conf. on Dependable Systems and Networks (DSN), Washington DC, USA.

COULOURIS, G., and DOLLIMORE, J. 1994. Security Requirements for Cooperative Work: A Model and its System Implications. In Proceedings of Workshop on ACM SIGOPS European Workshop: Matching Operating Systems to Application Needs. Wadern, Germany.

DEMCHENKO, Y., 2004. Virtual organisations in computer grids and identity management. Information Security Technical Report, 9, 1, 59-76.

EMMERICH, W., BUTCHART, L., CHEN, L., WASSERMANN, B., and PRICE, S. L., 2005. Grid Service Orchestration using the Business Process Execution Language (BPEL). UCL-CS. Research Note RN/05/07. Gower St, London WC1E 6BT, UK.

ERL, T. 2004. Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, Prentice Hall PTR, 0131428985

FISLER, K., KRISHNAMURTHI, S., MEYEROVICH, L.A. and TSCHANTZ, M.C. 2005. Verification and Change-Impact Analysis of Access-Control Policies. In Proceedings of the 27th International Conference on Software Engineering, 21, 196-205, St. Louis, MO, USA.

GREENBERG, M.M., MARKS, C., MEYEROVICH, L.A. and TSCHANTZ, M.C. 2005. The Soundness and Completeness of Margrave with Respect to a Subset of XACML. Technical Report CS-05-05, Department of Computer Science, Brown University.

HOLZMANN, G.J. 1991. *Design and Validation of Computer Protocols*, Prentice Hall.

HOLZMANN, G.J. 2004. T*he SPIN Model Checker, Primer and Reference Manual*, Prentice Hall.

JMS. 2001. *JMS: Java Messaging Service Specification (JMS)*, http://java.sun.com/products/jms/docs.html.

KREMER, S., MARKOWITCH, O. and ZHOU, J. 2002. An Intensive Survey of Fair Non-repudiation Protocols, Computer Communications, 25, 1601–1621.

KRISHNA, A., TAN, V., LAWLEY, R., MILES, S. and MOREAU, L. 2003. The myGrid Notification Service. In Proceedings of UK OST e-Science All Hands Meeting (AHM'03), Nottingham, UK.

LIBERTY. 2005. *Specification documentation of Liberty Alliance Project.* https://www.projectliberty.org/resources/specifications.php.

MARKOWITCH, O., GOLLMANN, D. and KREMER, S. 2002. On Fairness in Exchange Protocols. In Proceedings of 5th International Conference on Information Security and Cryptology (ISISC 2002), Springer LNCS 2587.

MOLINA-JIMENEZ, C., SHRIVASTAVA, S., CROWCROFT, J. and GEVROS, P. 2004. On the Monitoring of Contractual Service Level Agreements. In Proceedings of IEEE International Conference on E-Commerce (CEC), 1st International Workshop on Electronic Contracting (WEC), San Diego.

MOLINA-JIMENEZ, C., SHRIVASTAVA, S., SOLAIMAN, E. and WARNE, J. 2003. Contract Representation for Run-time Monitoring and Enforcement. In Proceedings of IEEE International Conference On E-Commerce (CEC), Newport Beach, USA.

MOLINA-JIMENEZ, C., SHRIVASTAVA, S., SOLAIMAN, E. and WARNE, J. 2005. A Method for Specifying Contract Mediated Interactions. In Proceedings of 9th IEEE International Enterprise Computing Conference (EDOC), Enschede, Netherlands.

MORGAN, R.L., CANTOR, S., CARMODY, S.,

HOEHN, W. and KLINGENSTEIN K. 2004. Federated Security: The Shibboleth Approach. Educause Quarterly, 27, 4.

MS Passport 2005, http://www.passport.net

NORFOLK, D. 1995. The Virtual Enterprise, Information Age, November, 32-39.

OASIS. 2003. eXtensible Access Control Markup Language (XACML) Version 1.0. OASIS Standard, http://www.oasis-open.org/committees/xacml.

OASIS. 2004. *Security Assertion Markup Language (SAML) v2.0*. http://www.oasis-open.org/committees/security.

PALLICKARA, S. and FOX, G. 2003. NaradaBrokering: A Distributed Middleware Framework and Architecture for Enabling Durable Peer-to- Peer Grids. In Proceedings of ACM/IFIP/USENIX Int. Middleware Conf., Rio de Janeiro, Brazil.

PERIORELLIS, P., TOWNSON, C.J.W., DUNNING-LEWIS, P. and ENGLISH, P.J. 2004. Draft GOLD Requirements Document v1.0, *Technical Report 854*, School of Computing Science, University of Newcastle upon Tyne.

PERRIN, T., ANDIVAHIS, D., CRUELLAS, J.C., HIRSCH, F., KASSELMAN, P., KUEHNE, A., MESSING, J., MOSES, T., POPE, N., SALZ, R. and SHALLOW, E. 2003. Digital Signature Service Core Protocols and Elements. OASIS Committee Working Draft, http://www.oasis-open.org/committees/dss.

ROBINSON, P., COOK, N. and SHRIVASTAVA, S. 2005. Implementing Fair Non-repudiable Interactions with Web Services. In Proceedings of 9th IEEE International Enterprise Computing Conference (EDOC), Enschede, Netherlands.

ROSHAN, K.T. and SANDHU, R.S., 1997. Task-Based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-Oriented Authorization Management. In Proceedings of IFIP TC11 WG11.3 Eleventh International Conference on Database Securty XI: Status and Prospects.

SANDHU, R.S., COYNE, E.J., FEINSTEIN, H.L. and YOUMAN, C.E. 1996. Role-Based Access Control Models, IEEE Computer, 29, 38-47.

SHARMAN, N., ALPDEMIR, N., FERRIS, J., GREENWOOD, M., LI, P. and WROE, C. 2004. The myGrid Information Model. In Proceedings of the UK e-Science All Hands Meeting 2004, 1 September.

SKONNARD, A. 2002. The XML Files: The birth of Web Services, MSDN Magazine, 17, 10.

SMITH R. 2005. Defining Virtual Organisations. *Technical Report 965*, School of Computing Science, University of Newcastle upon Tyne.

SOLAIMAN, E., MOLINA-JIMENEZ, C. and SHRIVASTAVA, S. 2003. Model Checking Correctness Properties of Electronic Contracts. In Proceedings of International Conference on Service Oriented Computing (ICSOC), Springer LNCS 2910, Trento, Italy.

THOMAS, R. K. 1997. Team-based Access Control (TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments. In Proceedings of Second ACM Workshop on Role-based Access Control, Fairfax, Virginia, United States.

WS-EVENTING. 2004. Web Services Eventing Specification (WS-Eventing), http://www-128.ibm.com/developerworks/webservices/library/specification/ws-eventing/

WS-NOTIFICATION. 2005. Web Services Notification Draft Specifications (WS-Notification), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn

WS-RELIABLEMESSAGING. 2005. *Web Services Reliable Messaging Protocol (WS-ReliableMessaging)*. http://msdn.microsoft.com/ws/2005/02/ws-reliablemessaging/.

WS-RELIABILITY. 2004. Web Services Reliable Messaging TC WS-Reliability 1.1. OASIS Committee Working Draft, http://www.oasis-open.org/committees/wsrm/.

WS-SECURITY. 2004. Web Services Security: SOAP Message Security 1.0 (WS-Security 2004). OASIS Standard 200401, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf.

WS-TRUST. 2004. *Web Services Trust Language (WS-Trust)*. http://msdn.microsoft.com/ws/2004/04/ws-trust/.

WU, J. and PERIORELLIS, P. 2005a. Authorization-Authentication Using XACML and SAML. *Technical Report 907*, School of Computing Science, University of Newcastle upon Tyne.

WU, J. and PERIORELLIS, P. 2005b. Evaluation of autorisation-Authentication tools. *Technical Report 935*, School of Computing Science, University of Newcastle upon Tyne.

XKMS. 2005. XML Key Management Specification (XKMS 2.0). W3C Recommendation, http://www.w3.org/TR/xkms2/.