

# Contract-Mediated Interorganizational Interactions

Santosh Shrivastava, *Newcastle University*

**A mediation service for monitoring terms of service can facilitate collaboration in virtual organizations by ensuring secure, private access to service resources.**

Opportunities abound for combining the Internet's many exciting services and resources into new, value-added interorganizational services. Increasingly, such collaborative ventures, or *virtual organizations*, will encourage strategic alliances among groups of organizations that share services electronically to satisfy mutual business goals. These arrangements allow each organization to maintain its autonomy except for the alliance's agreed-on undertakings.

VO operations management requires regulated access to service resources so that organization-specific resource-sharing policies occur securely and with integrity—a difficult situation because each potentially accessible organization might not unconditionally trust the others. Accordingly, all organizations in a VO will require strictly controlled and policed interactions. So, business process relationships will require protected trust management procedures for *terms and conditions*—rules that typically govern a conventional business partnership laid down in a paper-based contract (see the related sidebar)—and *quality of service*.<sup>1-3</sup>

So, we envision a VO requiring a mediation service for monitoring and enforcing terms and conditions as well as a QoS-monitoring service. One or more trusted third parties (TTPs) could provide these two services. Indeed, legal regulatory requirements might explicitly state that third parties must provide such services. At Newcastle University, my distributed systems research group has addressed several design issues for the mediation service.

## Conversations

A VO supported by protected trust management procedures needs electronic representations of its terms and conditions contracts. This lets a service mediate the rights and obligations that each interacting entity promises to honor. In the worst case, the mediation service would detect and notify all interested parties of violations of agreed interactions. For this, the service would need to maintain a nonrepudiable audit trail of all interactions. To do this, the organizations taking part in the VO would need to convert the original natural language terms-and-conditions contract written by their lawyers and other nontechnical people into a machine-interpretable specification that they can use to mediate the business conversations. We define a *conversation* as a small business activity executed between two or more business partners to perform a well-defined task such as issue a purchase order or reimbursement, process payment, or cancel a purchase order. In our group, we looked at creating conversation specifications by carefully studying rights, obligations, and prohibitions in contract clauses written in a natural language.

You can abstract business contracts as a set of permissions (P), obligations (O), and prohibitions (F) that *actors* (also called agents or roles) must fulfill to benefit others by performing (or not performing) *actions* (also called operations). We define a *permission* as an action that an actor (say, a buyer or a seller) can perform if desired.

For instance, "The buyer can use his discretion to send a purchase order to the seller" is a buyer's permission that benefits the seller. Likewise, we define an *obligation* as an action that an actor is expected to perform; an example of a seller's obligation for the benefit of the buyer is "The seller is obliged to respond to the buyer within three days after receiving the purchase order." A *prohibition* is an action that an actor should not perform. An example of a seller's prohibition for the benefit of the buyer is "The seller shall not send offers to the buyer unless they are requested."

Executing a permission operation is optional in the sense that an actor doesn't incur penalties for not executing it; conversely, failing to execute an obligation operation or daring to execute a prohibited operation are contract violations, subjecting the offending actor to a possible *sanction*. A sanction can take different forms—for instance, it can grant the offended actor a permission (for example, the permission to charge 10 percent on top of the original price), it could refuse the offending actor a permission, or it could assign the offending actor a new obligation (for example, paying a fine). Figure 1 shows a small, hypothetical business contract that stipulates business action interactions between a buyer and a seller for the purchase of goods.

## **1. Offer to buy**

1.1 The buyer may use his discretion to send a purchase order to the seller.

1.2 The seller must confirm acceptance or rejection of the purchase order within 24 hrs of receiving the purchase order.

## **2. Payment**

2.1 The seller must send an invoice to the buyer within 7 days of accepting the purchase order.

## **3. Invalid messages**

3.1 The buyer and the seller are forbidden to send invalid messages.

## **4. Sanction**

4.1 Failures to honor obligations and prohibitions will result in fines equal to 20 percent of the item's cost. The offended party shall be granted permission to issue an invoice notification to the offending party.

4.2 Failure to respond to a fine shall be sorted out outside this contract.

## **5. Synchronization and transaction failure handling**

5.1 Should the buyer, the seller, or both detect a technical failure that prevents them from continuing the normal course of a transaction, they must send a failure notification message by any other means.

Figure 1. A hypothetical business contract stipulating business action interactions.

Table 1 lists the permissions, obligations, and prohibitions that comprise the contract. The number after P, O, and F is the number of the clause in the contract from which the permission, obligation, or prohibition originated. In the contract, clause 3.1 specifies prohibitions for the buyer and for the seller. To distinguish between these two cases, we named them F3.1<sub>B</sub> and F3.1<sub>S</sub>. Similarly, P4.1<sub>B</sub> and P4.1<sub>S</sub> stand for permissions for the buyer and the seller extracted from clause 4.1.

Contract permissions, obligations, prohibitions, and sanctions.

Permissions	Subject	Beneficiary	Sanctions
P1.1 Send purchase order.	Buyer	Seller	None
P4.1 <sub>B</sub> Issue invoice to fine.	Buyer	Seller	None
P4.1 <sub>S</sub> Issue invoice to fine.	Seller	Buyer	None
<b>Obligations</b>			
O1.2 Send confirmation within 24 hrs.	Seller	Buyer	P4.1 <sub>B</sub>
O2.1 Send invoice within seven days.	Seller	Buyer	P4.1 <sub>B</sub>
<b>Prohibitions</b>			
F3.1 <sub>B</sub> Send invalid messages.	Buyer	Seller	P4.1 <sub>S</sub>
F3.1 <sub>S</sub> Send invalid messages.	Seller	Buyer	P4.1 <sub>B</sub>

As it stands, this contract might not have enough details for the technical people in charge of creating its executable version, yet it contains important information for this stage. For instance, it has enough information to begin reasoning about the contract's correctness.

We can further refine the description in figure 1 to include implementation-specific technical details such as acknowledgments and synchronization messages. To show what an implementation-oriented contract looks like, we'll assume that the seller and the buyer have agreed to use the widely adopted RosettaNet process specification standard.<sup>4</sup> In RosettaNet, a buyer must use the Request Purchase Order *partner interface process*, PIP 3A4, to express a desire to buy (see figure 2). The seller must use the Notification of Invoice PIP (PIP 3C3) to invoice the buyer. The specification of such PIPs includes sending both business action messages and business signal messages. Recipients of a business action message must acknowledge it by sending a business signal message back.

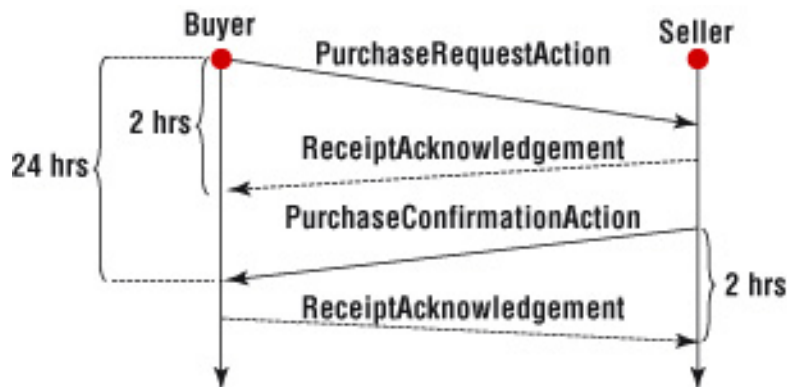


Figure 2. Conversation regarding a request for a purchase order.

Figure 3 shows parts of the modified English text version of the implementation-oriented contract. This version is different from the original one in that, it includes (in addition to the business actions messages) business signal messages to help the two business partners synchronize their interactions. The new clauses appear in bold font.

## 1. Offer to buy

1.1 The buyer may use his discretion to send a purchase order to the seller.

**1.2 The seller is obliged to acknowledge the purchase order within 2 hrs of receiving the purchase order.**

1.3 The seller is obliged to confirm if the purchase order is accepted or rejected, within 24 hrs of receiving the purchase order.

**1.4 The buyer is obliged to acknowledge the purchase order confirmation action within 2 hrs of receiving the message.**

## 2. Payment

...

Figure 3. A modification of the contract in figure 1 to include (in bold) business signal messages for synchronization.

In other work, we describe how you can convert contract specifications into finite state machine representations and check and their correctness properties.<sup>5,6</sup> The World Wide Web Consortium is developing a standard on Web service choreography, WS-CDL, that defines a language for specifying conversations.<sup>7</sup>

## Mediation service

Conceptually speaking, a mediation service sits between business partners so it can observe their business interactions. Each enterprise expects access to others' services, but the mediation service allows an operation to take place only if the contract's rules permit it and then only if invoked by a legitimate role within a participating enterprise. So, the mediator performs *access control* by intercepting all contractual operations that parties might try to perform. Our approach represents conversations as finite state machines and uses role-based access control for authenticated access. The SDS (state-dependent security decision) approach addresses relevant design issues. Each enterprise is autonomously responsible for its own role management and role assignments, thereby ensuring that each enterprise controls its own role management and role assignments. We assume that different roles will have the

permissions, obligations, and prohibitions to send messages of different types (requesting purchase orders, invoice notifications, payments, and so on.).

Our approach allows either *centralized* (see figure 4a), where for illustrative purposes we assume an interaction between buyer and seller), or *distributed* (see figure 4b) deployment. In a centralized deployment, a single TTP deploys the service. A single state machine represents a given conversation; an incoming message is checked for its role as well as associated permissions and obligations. If these are correct, it forwards the message to its final destination; it drops incorrect ones. Figure 5 shows the centralized mediated version of the purchase order conversation. Once deployed, the mediation service will guarantee that only *legal* messages (right type, sequence, and time) reach their final destination. Participants can trust incoming messages as correct and act on them with the guarantee that the mediator has already approved them; furthermore, it guarantees applications that illegal messages sent accidentally will never reach their counterparts.

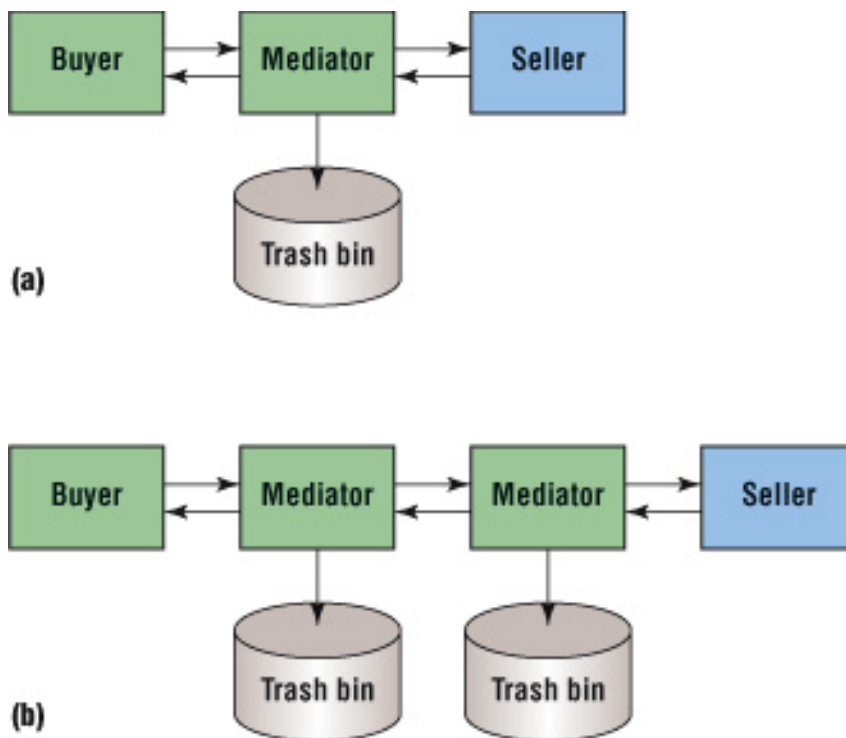


Figure 4. (a) Centralized and (b) decentralized deployment.

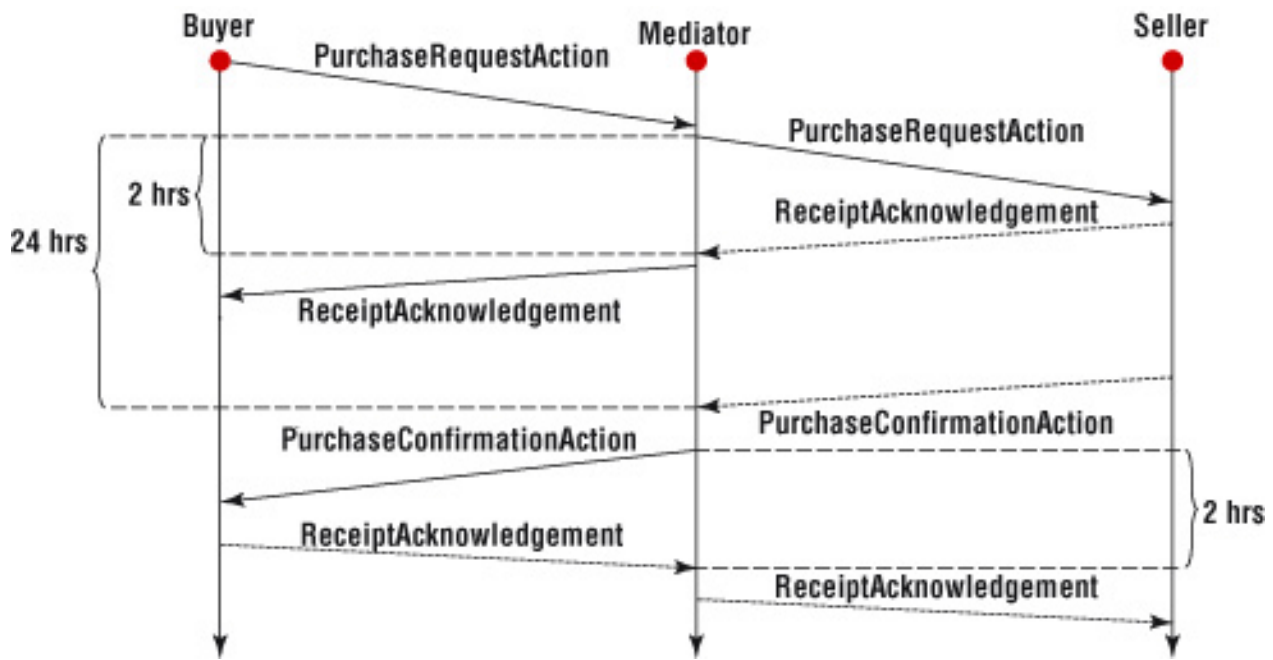


Figure 5. Mediated request purchase order conversation.

In a distributed deployment, the mediation functionality splits, with each side implementing one side of the conversation state machine. Figure 4 shows just two of several deployment scenarios. Determining which particular model suits a given VO setting poses an interesting research problem worthy of further investigation. Distributed deployments also face the challenge of keeping contract state information synchronized with all the mediators. For example, a valid message forwarded by the buyer's side could be dropped at the seller's end because intervening communication delays render the message untimely (and therefore invalid) at the seller side. State synchronization is necessary to ensure that both parties either agree to treat the message as valid or invalid.

The story doesn't end here. The mediation service must also provide facilities for *nonrepudiated interactions*, which is a research topic in its own right, and so my discussion here will be brief.

To support nonrepudiated interactions for regulating interactions, a given action must be attributable to the party who performed the action and commitments made must be attributable to the committing party. For example, a client shouldn't be able to subsequently disavow the request or a service's consumption. So, to regulate an interaction, we require action *attribution*, *validation*, and *auditing* of the parties involved. Nonrepudiable attribution binds an action to the party performing the action. Validation determines an action's legality with respect to interaction agreements. Auditing ensures that evidence is available in case of dispute and to inform subsequent interactions.

For example, to deliver a message from A to B:

- B may require nonrepudiation of origin (NRO) of the message—irrefutable evidence that the message originated at A, and
- A may require nonrepudiation of receipt (NRR) of the message—irrefutable evidence that B received the message. Nonrepudiation is usually achieved using public key cryptography.

If A signs a message with its private key, B can confirm the message's origin by verifying the signature using A's public key. Similarly, given B's signature on the message, A can confirm receipt by verifying the signature using B's public key. To support the assertion that a key used to sign evidence wasn't compromised at time of use and for audit trail logs, a mutually trusted third-party timestamping service should timestamp the signed evidence. Elsewhere, we describe how you can use component middleware to implement nonrepudiation.<sup>9</sup>

## Conclusion

Our work described how to mediate the rights and obligations to services for simple contracts. However, contracts can be quite complicated and much research is under way on contract representation.<sup>10-12</sup>

## Acknowledgments

The UK Engineering and Physical Sciences Research Council partly funded this work under e-Science program grants, Trusted Coordination in Dynamic Virtual Organisations and Grid-Based Information Models to Support the Rapid Innovation of New High Value Added Chemicals (GOLD). The European Union also partly funded this work under projects IST-2001-34069: Tapas (trusted and qos-aware provision of application services) and IST-2001-37126: Adapt (middleware technologies for adaptive and composable distributed components).

## References

1. C. Molina-Jimenez et al., "On the Monitoring of Contractual Service Level Agreements," *Proc. 1st IEEE Int'l Workshop Electronic Contracting (WEC 04)*, IEEE

Press, 2004, pp. 1-8.

2. A. Dan et al., "Web Services on Demand: WSLA-Driven Automated Management," *IBM Systems J.*, vol. 43, no. 1, 2004, pp. 136-158.
3. H. Ludwig, A. Dan, and R. Kearney, "Cremona: An Architecture for Creation and Monitoring of WS-Agreements," *Proc. Int'l Conf. Service Oriented Computing (ICSOC 04)*, ACM Press, 2004, pp. 65-74.
4. *RosettaNet Implementation Framework Core Specification*, v. 2, RosettaNet, July 2001.
5. E. Solaiman, C. Molina-Jimenez, and S. Shrivastava , "Model Checking Correctness Properties of Electronic Contracts," *Proc. Int'l Conf. Service Oriented Computing (ICSOC 03)*, LNCS 2910, Springer, 2003, pp. 303-318.
6. C. Molina-Jimenez et al., "Run-Time Monitoring and Enforcement of Electronic Contracts," *Electronic Commerce Research and Applications*, vol. 3, no. 2, 2004, pp. 108-125.
7. *Web Service Choreography Description Language*, <http://www.w3.org/TR/ws-cdl-10v.1.0>, World Wide Web Consortium (W3C), Dec. 2004.
8. J. Biskup, T. Leineweber, and J. Parthe , "Administration Rights in the SDS System," *Data and Applications Security XVII Status and Prospects*, S. De Capitani di Vimercati and I. Ray, eds., Kluwer, 2004, pp. 149-162.
9. N. Cook, P. Robinson, and S. Shrivastava, "Component Middleware to Support Nonrepudiable Service Interactions," <http://doi.ieeecomputersociety.org/10.1109/DSN.2004.1311931>, *IEEE/IFIP Int'l Conf. Dependable Systems and Networks (DSN 04)*, IEEE CS Press, 2004, pp. 605-614.
10. O. Perrin and C. Godard, "An Approach to Implement Contracts as Trusted Intermediaries," *Proc. 1st IEEE Int'l Workshop Electronic Contracting (WEC 04)*, IEEE CS Press, 2004.
11. O. Marjanovic and Z. Milosevic , "Towards Formal Modelling of e-Contracts," *Proc. IEEE Int'l Enterprise Distributed Object Computing Conf. (EDOC 01)*, IEEE CS Press, 2001.
12. C. Molina-Jimenez , S. Shrivastava, and J. Warne, , "A Method for Specifying Contract Mediated Interactions," *Proc. IEEE Int'l Enterprise Distributed Object Computing Conf. (EDOC 05)*, IEEE CS Press, 2005, pp. 106-115.



**Santosh Kumar Shrivastava** is a professor of computing science at the University of

Newcastle upon Tyne, where he leads the Distributed Systems Research Group. His research interests include distributed computing, spanning middleware, fault tolerance, and applying transactional middleware technologies to contract management in virtual organizations. He received his PhD in computing science from Cambridge University. Contact him at School of Computing Science, Univ. of Newcastle, Newcastle upon Tyne, NE1 7RU, UK; santosh.shrivastava@ncl.ac.uk.

## Terms and Conditions

A partner in a VO providing a service to other partners will need several assurances—for example, that the service invoker can invoke the operation or has been authenticated as well as evidence of maintained interaction (*nonrepudiation*). Service consumers will need complementary assurances that we call *terms and conditions monitoring and enforcement*. Terms and conditions express what operations or actions the business partner is permitted, obliged, and prohibited to execute. Additionally, the rules stipulate when and in what order to execute the operations. For instance, for a buyer-seller business partnership, the contract will stipulate when the buyer must submit purchase orders and within how many days of receiving the purchase order the seller must deliver the goods, and so on.

In addition to terms and conditions, providers and consumers also need *service-level agreements* stating the quality of service, such as availability and response time. For example, within a business-to-business auction, the auctioneer might need to guarantee that "even during peak periods, invoking the `place_bid` operation will complete successfully within two seconds when fewer than 100 bidders are logged in." For most services, any degradation by the consumer in the perceived QoS level can have serious negative consequences. Providers should ensure that the offered service meets the agreed QoS. Contractual SLAs should specify the QoS level delivered to the consumer. As the name suggests, monitoring contractual SLAs involves collecting statistical metrics about a service's performance to evaluate whether the provider complies with the expected QoS level.

## Related Links

DS Online's Middleware Community, <http://dsonline.computer.org/portal/site/dsonline/>

"Designing Interaction Systems for Distributed Applications",  
<http://dsonline.computer.org/portal/site/dsonline/>

"Programming Pervasive Spaces", <http://doi.ieeecomputersociety.org/10.1109/MPRV.2005.22>

**Cite this article:** Santosh Shrivastava, "Contract-Mediated Interorganizational Interactions," *IEEE Distributed Systems Online*, vol. 6, no. 11, 2005.