Online Dispute Resolution Systems as Web Services

Bonnet, Vincent; Boudaoud, Karima; Gagnebin, M.; Harms, Jurgen; Schultz, Thomas

Online Dispute Resolution Systems as Web Services

Proceedings Hewlett-Packard OpenView University Association Workshop
Held on videoconference, workshop on June 11-13 2002
Available online http://www.hpovua.org/publications/proceedings

V. Bonnet[1], K. Boudaoud[1], M. Gagnebin[1], J. Harms[1] and T. Schultz[2]

[1] CUI - University of Geneva
Email: {vincent.bonnet, karima.boudaoud,juergen.harms@cui.unige.ch}

[2] Faculty of Law, University of Geneva
Email: thomas.schultz@droit.unige.ch

1 Introduction

With the growth of information technology and the Internet, a new always-on and global marketplace has transformed business: electronic commerce has appeared in the business and legal landscape. Inevitably, with this new form of commerce came new forms of disputes. As transactions in a global market mean an increased probability of transnational disputes, parties situated sometimes on different continents are opposed over small claims. Courts or traditional out-of-court dispute resolution mechanisms cannot reasonably resolve such conflicts. As a consequence, a new tool for dispute resolution has appeared, which is more efficient, more cost effective and more flexible than traditional approaches: this is Online Dispute Resolution (ODR). The parties to a deal that has gone awry are offered the possibility to solve their dispute over the Internet, communicating by means of emails, chat-rooms, videoconferences and other electronic means. Although ODR is a booming business, with some institutions reporting having handled over 200'000 disputes in a little over two years, many technical issues have to be addressed to increase the quality and reliability of "online justice". For lawyers, solving a dispute means reconstructing what has happened, in order to determine who is right and who is wrong. In cyberspace, this can be difficult.

ODR systems, a very specific kind of Web services, require the support of new types of communication adapted to dispute resolution processes (negotiation, mediation, and arbitration) and corresponding security mechanisms such as integrity and confidentiality of sensible data. The goal of this paper is to analyze the principle characteristics that an ODR system must fulfill, mapping the legal requirements to a structure of technical concepts. The adequacy of these concepts is discussed in the light of a series of questions flowing from common legal schemes of dispute resolution. In addition, we propose in this paper a modeling of an ODR service by identifying the different actors and the communication types they use.

Our paper is organized as follows: first, we give an overview of ODR systems; then, we discuss some technical and particularly security requirements for ODR; finally, we describe the modeling of an ODR service based on a simple scenario of a dispute resolution.

2 Overview of ODR Systems

The term ODR characterizes new methods of dispute resolution the major part of which is provided online. Most ODR services are alternatives to litigation and to state justice. In this sense, they are the online transposition of the methods developed in the Alternative Dispute Resolution (ADR) movement, which are mainly negotiation, mediation, and arbitration. But there are also projects of proper online courts, which are really normal court which simply communicate essentially online. There are four main forms of ODR systems: automated negotiation, assisted negotiation, mediation, and arbitration.

In automated negotiation (also called blind-bidding) the parties successively submit to a computer a monetary figure as settlement proposal, the computer then compares the offer and the demand and reaches a settlement for their arithmetic mean. Almost 20 providers offer it and some of them handle up to 3'000 disputes a month

In assisted negotiation, the parties communicate with one another over the Internet, using for instance emails, web-based communication tools or videoconferences. The providers offer storage means and secure sites. Assisted negotiation is extremely successful, with over 20 providers and up to 20'000 disputes solved each month.

Online mediation is the online form of traditional mediation. A third neutral person with no decision power tries to convince the parties to reach an agreement. The only difference with offline mediation is that the third neutral and the parties always communicate via the Internet. Although there are many ODR providers which offer online mediation, only few cases are solved by such a process, probably because such a system is technologically difficult to set up, as the parties usually ask for highly developed communication means.

Online arbitration is similar to traditional arbitration, in the sense that a third party chosen by the parties, or nominated by the institution chosen by the parties, renders a decision on the case after having heard the relevant arguments and seen the appropriate evidence. Arbitration decisions are not yet recognized nor enforced in the legal systems, but self-enforcement structures are being worked on, for instance by instructing a private company to transfer a domain name, by threatening to remove a trustmark, or by controlling a fund through which all payments to traders transit. In online arbitration, the parties usually communicate by emails, web-based communication tools and videoconferences. There are more than 25 ODR providers which offer online arbitration, but it is mostly successful in the filed of domain names **[1]**.

Other less deployed types of ODR exist, such as cybercourt, the first of which is due to begin activities in October of this year, or online mock-trials, in which the parties can reality-test their case before it comes to litigation. Providers of ODR also offer additional services, such as complaint assistance, which consists of support in search for counsel, forwarding complaints to trustmarked traders or calling on them to take action. Other providers offer services of dispute prevention, which consists for instance of checks of employees prior to employment, standard business contracts and forms, and training of employees and employers. Legal literature or portals to other services are also often provided.

3 Online evidence in online dispute resolution

For lawyers, solving a dispute means reconstructing what has happened, in order to determine who is right and who is wrong. With ODR, this raises many issues **[1]**. First, the author of each message related to the conclusion of the contract must be identified. Second, the messages containing the offer, the acceptance and the general conditions of the contract must be submitted to the dispute resolver. Third, evidence may have to be provided that a file or program has been entirely transmitted to the buyer. Fourth, the dispute resolver must be convinced of the identity of the author of the messages that are sent during the ODR procedure. Fifth, when gathering evidence from electronic repositories, is must be ascertained that that contents of the repository have remained unchanged they were stored. Sixth, the information regarding evidence that has been sent to the dispute resolver must be protected against third parties. Seventh, after the dispute resolution procedure, the decision has to be notified to the parties without them being able to repudiate the notification. Finally, when the decision must be sent to an enforcement authority, by it public or private, the authority must be able to authenticate the original and the risk of manipulation must be satisfactorily excluded.

Under most laws, the rules of evidence are sufficiently flexible so as to allow the dispute resolver to give electronic data as much probatory force as he or she thinks they deserve, as long as this assessment by the dispute resolver is not arbitrary. The goal is therefore to provide technical solutions which convince a dispute resolver of the authentic character of a piece of evidence. In most cases, this means that the security of the data and its transmission must be ensured.

4 Technical and Security Issues

ODR as legal issue requires answering many questions. Those questions reveal the legal requirements of ODR Systems and Infrastructures in which they are involved. We point out that a coherent and adequate ODR System (ODRS) must fulfill principle characteristics, mapping those legal requirements to a structure of technical concepts.

4.1 From Legal Evidence To Security Concepts

In this part we focus on security requirements. In fact, the legal requirements listed previously can be concretized by more pragmatic questions (and their short but theoretical answer in classic computer science):

- Can we make the proof of the content of a valid contract and to allow the transmission of its content to an ODR for the evaluation of this proof? *(Yes)*

- Can we give the proof that a document was completely sent? *(Yes)*
- Can we ensure the integrity of submitted information? *(Yes)*
- Can we protect information stored on an ODR database from unauthorized party? *(Yes)*
- Would it be possible to create a notification model allowing the proof of delivery without the intervention of receiver? *(No)*
- Can we identify the sender of a message during the conclusion of the contract and the ODR procedure? *(Yes)*
- Can we distinguish an original from a copy? (And what could be original information in computer science?)

4.1.1 Communication vs data management

In our work we argue that all the requirements based on evidence can lead to technical security concepts.

- On the one hand, ODRS requires to take care of communication channels between the actors of all systems. Each channel must be isolated to prevent interception of message and data integrity, if needed.
- On the other hand, ODRS requirements are based on the notion of deeper evidence in the light of data management. Here, the data is potentially or already an evidence:

  ✓ "Potentially" because a data can be communicated in order to help dispute resolution preventively.

  ✓ "Already" because a data can be transmitted with the objective to be a legal evidence in a specific case. Besides, the second type of data can later be involved in another dispute if the primary dispute resolution (where it was playing the role of evidence) is disputed itself. Thus, it is again potential evidence. The potential use of a data as evidence could require us to admit any communicated data can be involved in the resolution mechanism.

We distinguish data and communication channel because a communication channel is not exactly the sum of data communicated during the time of channel opening. There are three major reasons to this distinction:

- A message is not a data. A data have no destination, source of emission, nor exchange goal by construction.
- The absence of message is sometimes a precious information. Furthermore, the construction of a communication channel between too agents is already a precious information. The simple fact to be involved in a VPN (Virtual Private Network) without any contribution to the network is information. This is more and more true with the covered-channels where it is impossible to measure distance between communicant and non-communicant parties.
- The related contextual information to a channel communication is a shared context of information on a channel. The channel is seen as composed of service interface.

Our objective would be to have a very Data/Information Centric approach but we must take care of Communication channel and Message at the very early source of our legal problematic. We got by those ways some of issues discussed in **[2]** but make them more accurate and well organized in the following.

So let us focus on the Non-repudiation property **[3] [4] [5],** by distinguishing Communication and Data and then Message and Data. We introduce also in our perspective the Non-repudiation properties concerning Data generation and retention. Beside, we add a primary definition of "responsibility condition" by extended Who, When, by a general "Responsibility Condition" element.

For instance:

> The classic definition of Non-repudiation approval*: "non-repudiation of approval service provides proof of whom is responsible for approval of the content of a message"*; becomes → Non-repudiation of

approval service provides proof under responsibility conditions for approval of the content of a message. The responsibility conditions are then: which agent support responsibility, which agent supports risk, when the responsibilities are taken for such information (An agent is here a generalization of Actor).

On this base, we present the *Non-repudiation Property* of a Message as a Service, and then of a more basic Data.

Non-repudiation Property/Service of a Message

Name	Non-repudiation means
Approval	Non-repudiation of approval service provides proof under responsibility conditions for approval of the content of a message
Sending	Non-repudiation of sending service provides proof under responsibility conditions of who send a message
Origin	Non-repudiation of origin service is a combination of approval and sending services
Submission	Non-repudiation of submission service provides proof under responsibility conditions that a delivery authority has accepted a message for transmission
Transport	Non-repudiation of transport service provides proof under responsibility conditions for the message originator that a delivery authority has given the message to the intended recipient
Receipt	Non-repudiation of receipt service provides proof under responsibility conditions that the recipient received a message
Knowledge	Non-repudiation of knowledge service provides proof under responsibility conditions that the recipient recognized the content of a received message
Delivery	Non-repudiation of delivery service is a combination of receipt and knowledge services as it provides proof under responsibility conditions that the recipient received and recognized the content of a message

Non-repudiation Property/Service of Data

Name	Non-repudiation means
Creation	Non-repudiation of creation service provides proof under responsibility conditions of whom is creator of the content of the data
Opening	Non-repudiation of service provides proof under responsibility conditions of the action of open and decrypts the data.
Destruction	Non-repudiation of destruction provides proof under responsibility conditions of destruction (end of life) of the data.
Emd-distribution	Non-repudiation of destruction provides proof under responsibility conditions of whom is the terminal diffuser of the data. The data cannot be resent in the content of a message to another party. Useful against screen-scrapers **[6]**
….	….

We argue that non-repudiation property is the primary effort to make in order to resolve any communication and data management problem. On basis of such property we then complete the technical structure response and discuss Integrity, Confidentiality and Availability.

4.1.2 Towards Integrity, Confidentiality, Availability

This "by-evidence or non-repudiate approach" reveals that:

1. By “strong non-repudiation protecting” the data (and thus the message), we could deal with any dispute, help the resolution mechanism, and also prevent the dispute on a dispute resolution. But, further, we could eliminate risks of dispute since we could prevent risks linked to data handling responsibilities.

2. Since the Integrity property of information is determinate by the fact that a data is really the data that it claims to be; this property is reduced to the non-repudiation of sender/creator/ property to be itself.

3. Since the Confidentiality of an information is determinate by the fact that unauthorized agents can take knowledge of the signification of the data; this property is reduced to the good exploitation of

cryptographic methods applied conjointly to safe procedures of verification of non-repudiation property on source, destination, creation… That is a step toward a correlated specific Service of verification.

4. The special case of Deny Of Service, which concern endpoint availability and then data potential mobility, would be the only point of security that could not be address in "by-evidence and non-repudiation approach". In fact, availability of justice is a very important concept. Anyway, it is dependent of the communication channel integrity.
5. Another aspect of the availability is the feedback availability. Make a sender aware of the sending and the receiving and the knowing of the content must take part of the system. The System must relay (as a real service) the non-repudiation attributes of a data or a message to agents (user or automated). This could be part of the notification model required.

What we have done is to build correspondence between data and communication property and announce service-oriented response. We have already started to define what could be an ODR Service and what should be Exchanged-Data.

4.2 ODR as a Specific Web Service

In this section, we want address the security aspect (by-evidence non-repudiation, integrity, confidentiality, availability) in a Web Service Oriented Framework. We focus mainly on the W3C Web Services framework.

There is a consensus on a set of technology and tools that can represent the Web-services optimism in increasing the commercial market of the Web: XML, SOAP, WSDL, UDDI **[7] [8] [9] [10] [11].**
The preliminary aims of those bricks were to reach the necessary interoperability property of over world multi-pole information system. But, this was not to address directly (or at a very early stage) any legal and worse, security requirements!

Still under the security concern, let us introduce and regard the four technologies involved and detail, each time, security issues.

4.2.1 The quartet of the W3C Web services technologies

4.2.1.1 XML

XML is a standard of communication between applications **[7].** Relatively to our subject, *XML Digital Signatures* complete the XML standard for verifying the origins of messages **[12]**. The XML signature specification allows XML documents to be signed in a standard way, with a variety of different digital signature algorithms. Digital signatures can be used for the validation of messages and for the non-repudiation.
Besides XML Encryption will allow encryption of digital content, such as Graphical Interchange Format (GIF) images, Scalable Vector Graphics (SVG) images, or XML fragments. XML Encryption allows the parts of an XML document to be encrypted while leaving other parts open, encryption of the XML itself, or the super-encryption of data (i.e., encrypting an XML document when some elements have already been encrypted). As part of the Java Community Process (JCP), there are two Java Specification Requests (JSRs) that are currently in progress; JSR105 XML Digital Signature 1.3 and JSR106 XML Digital Encryption.14 When complete, these two JSRs will define the standards in Java for each technology, thus standardizing the interfaces in each vendor's Web services toolkit.
Finally, ML Key Management Specification (XKMS) want to cover the registration and distribution of XML-based public keys to encrypt and decrypt documents **[13]**. This is part of the PKI management framework that is necessary to support such signing services.

If we confront this XML overview to ODR Systems, the XML completion by XML Digital Signature and XML Encryption tend to respond to part of non-repudiation problematic mixed with encryption-protected exchanged-data. This healthy structure is useful in ODR systems process for claim standard description (odrXML definition **[14]**) but also could be used to:

- describe ODR Process Workflow ;
- more important, to prepare exchanged-data to ODR (legal) efficient treatment by describing non-repudiation by signing on responsibility for exchanged-data both in contract and transaction agreement and in ODR procedure involvement.

4.2.1.2 SOAP

Simple Object Access Protocol is a XML-based mechanism for data exchange. It consists of three parts: the envelope that describes the framework to describe the content of a message, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls (RPC) and responses. SOAP can potentially be used in combination with a variety of other protocols (HTTP, HTTP Extension, RMI/IIOP, SMTP, FTP, MQ). **[8]**

SOAP 1.1 (latter version) did not include provisions for signing messages and thus lacks this security primitive. That's why the SOAP Security Extensions via Digital Signature (SOAP-DSIG), defines the syntax and processing rules for digitally signing SOAP messages and validating signatures. **[SOAP Security Extension].** However the simultaneously usage of SOAP-DSIG and SSL is not sufficient for Non-repudiation **[16]** since it is vulnerable to at least two ambiguous transactions:

- a receipt can claim to have received a message twice, or
- a sender can claim to have sent a message only once

Indeed the digital signature scheme guarantees nothing about how many time the message was signed and sent by the sender. The parade is the use of a nonce. A *nonce* is a non-repeating string freshly generated by the sender (the signer) such that the intended recipient can check its uniqueness. Typically, the nonce can be implemented as a counter (a sequence number) or as a timestamp.
Thus SOAP introduces a break in the non-repudiation property. So it is important for security against repudiation to add in SOAP message both a nonce and the identity of the intended recipient to application data to be signed.

4.2.1.3 WSDL

Web Service Description Language is an XML Interface Description Language (IDL-CORBA like). This interface is the endpoint reacting to message. A message in WSDL can be document-oriented or procedure-oriented. WSDL allows description of protocol used to communicate. It supports actually SOAP 1.1, HTTP GET/POST and MIME. **[9]**

4.2.1.4 UDDI

The WSDL service information can be extracted from a UDDI (Universal Description, Discovery, and Integration) Business service entry, or may be obtained from other service repository sources. **[10]**

WSDL and UDDI should address uniquely the availability property of such ODR Service. But yet consider that the service description and UDDI delivery (redirecting request of service toward good services) are part of information used in transaction. Thus they are part of "responsibility condition" (the context of service providing).

4.2.2 The overall Web services framework

If we take Web-Services Framework in order to describe the landscape of Web-Service effort, we must mention that the first industry standard for secure e-com is Security Assertion Markup Language **[Z16]**. SAML is being developed to provide a common language for sharing security services between companies engaged in transaction. SAML allows companies to securely exchange authentication, authorization, and profile information between their customers, partners, or suppliers regardless of their security systems or e-commerce platforms. As a result SAML promotes the interoperability between disparate security systems, providing the framework for secure e-business transactions across company boundaries. The SAML specification also provides for encrypting the SOAP message content itself when being transported over unencrypted links.

Considering the point of Service Management, Interoperability is a healthy base of manageability of such a service. However Security and Right to Dispute must be integrated into an efficient *Web Services Management Platform*. Web Services Management Platform can already be composed of XKMS implementation framework. One key benefit of signing action (related to responsibility engagement) is really the concept of non-repudiation. With signatures, service providers can not only provide evidence that a document is valid but also log the related message transactions into signed audit logs managed by part of the service. Once an audit log has been signed it cannot be modified without significantly changing the signature. Note that hackers often modify audit logs in order to "cover their tracks" to avoid detection. When third party non-repudiation is required, digital receipts provide independent verification that specific transactions have occurred. Signed log files help prevent such situations but signed log management would deal with an important amount of information. This information must be stored in efficient database (or repositories), to protect and to prepare for ODR Systems Service exploitation. Those problems has been already discussed in studies about non-repudiation services on CORBA Framework **[5]** and must be evaluated seriously in regarding the distance between W3C Web-service Framework and CORBA Framework.

Let us know start the process of translating ODR Systems in a Web-services by modeling ODR services and to expose the functionality of a valuable ODR Service.

5 Modeling of ODR services

In this section, we model an ODR system using UML formalism. We first identify the different actors implied in an ODR system and its functionalities. Then we represent the types of communication between the different actors in each ODR process (negotiation, mediation, arbitration). Finally, we describe a simple scenario of a dispute and its resolution.

5.1 Different actors

The different actors implied in ODR system are:

- parties
- arbitrator
- mediator
- experts
- witnesses
- ODR-administrator
- system administrator
- visitor

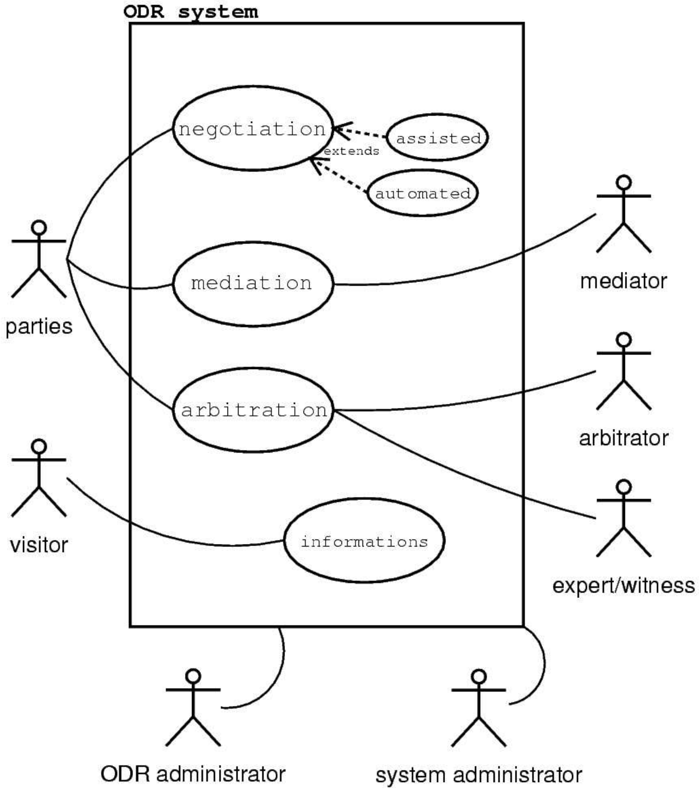


Figure 1 : Use case diagram

Parties, arbitrators, mediators, experts and witnesses are directly implied in ODR processes.

ODR-administrator and system administrator are indirectly implied in ODR process because they are not bound with any particular process but with the whole system.
Visitors are persons browsing the ODR web site to get information. They can be interested persons, potential users or even court representants, in the case or enforcement.

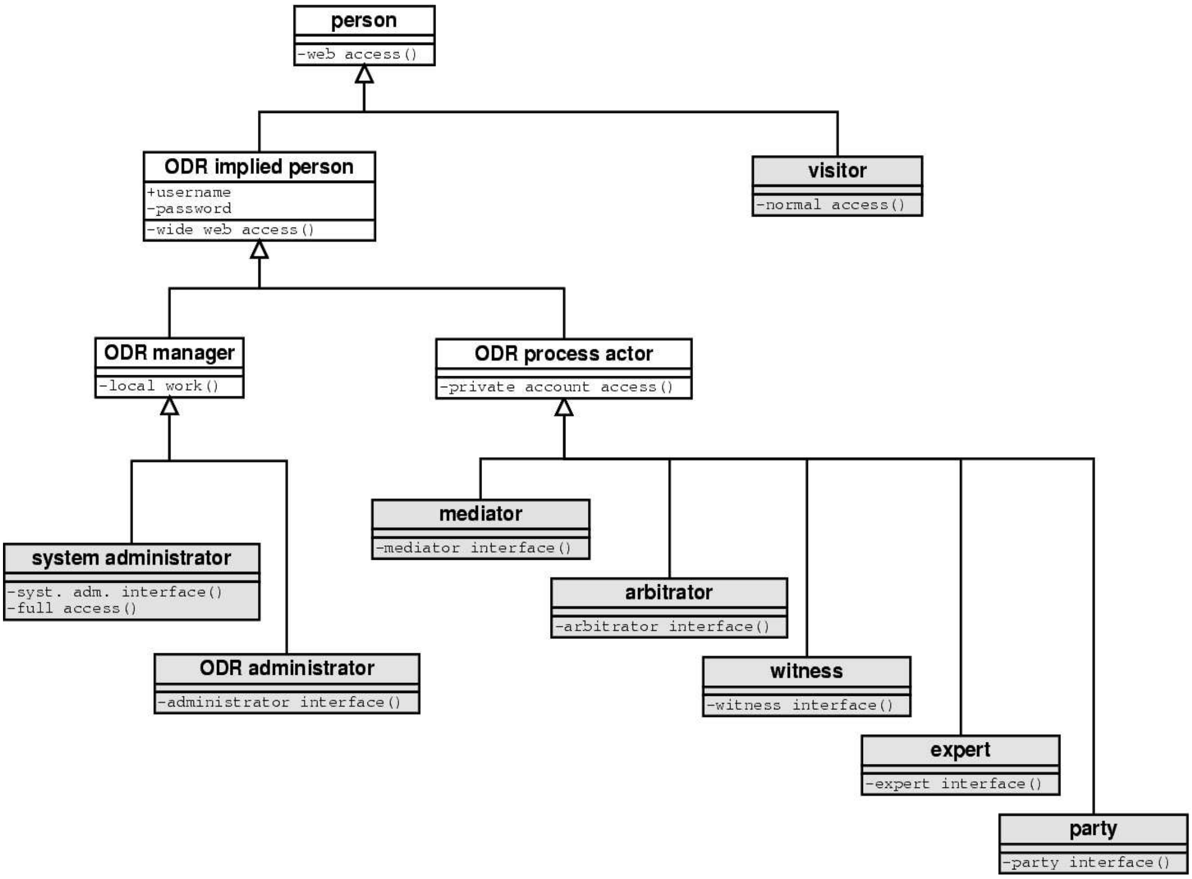


Figure 2 : Class diagram

We distinguish three kinds of actors:

- The actors implied directly in an ODR process:
  - ✓ parties in every process,
  - ✓ arbitrator, witnesses and experts in arbitration process,
  - ✓ mediator in mediation process.
- The actors managing the ODR system:
  - ✓ system administrator,
  - ✓ ODR-administrator.
- The visitors.

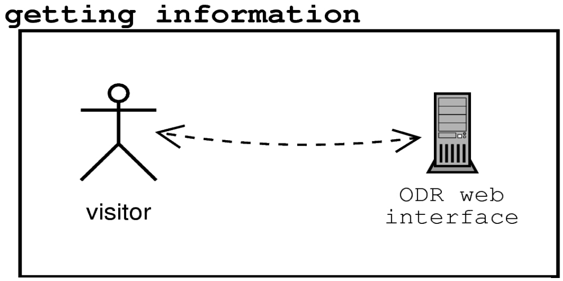
Each of these categories has different access rights to the system. Visitors can only access the public side of the web site. ODR implied persons have a private account in the ODR system they can access through a specific interface. The whole actor-side process is front-ended in this interface. ODR managing persons have of course a wider access to the system. The ODR administrator has a full access to legal aspects of the system, the system administrator, on the technical aspect.

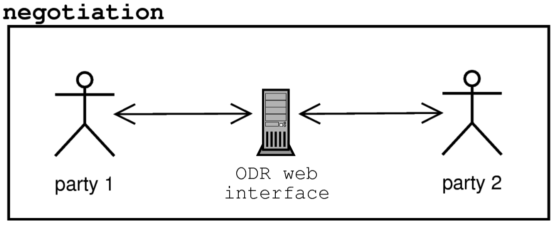
5.2 Types of Communication in ODR systems

The following four diagrams describe communications between actors relatively to the different ODR processes.

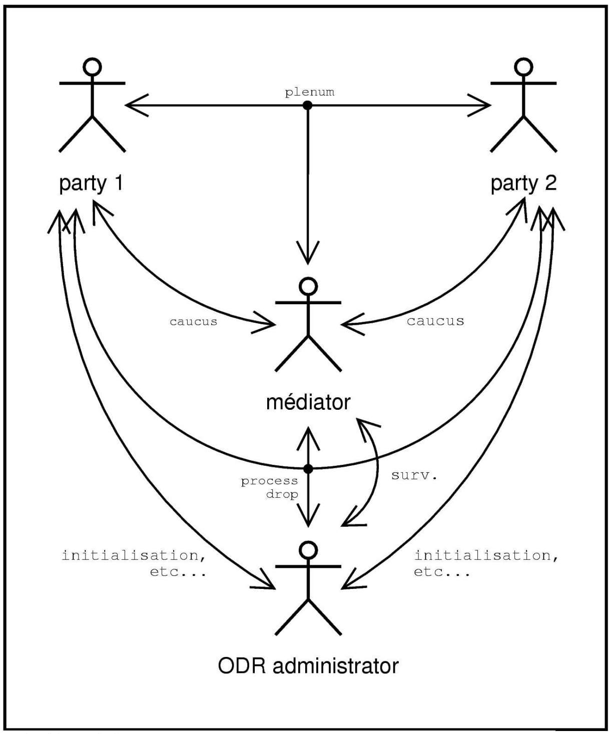
In case of negotiation, mediation and arbitration, each communication has to satisfy security requirements (discussed in last section), and has to be conserved in system archives. This is not necessary when the ODR service is accessed for getting information.

Communications represented as doted arrows are basic not secured communications. Communications linked to a black bullet are visible to every actor connected to this bullet through any communication link. This type of communication correspond to the legal concept of plenum. It must be secured. Other communications are private and secured actor to actor communications.

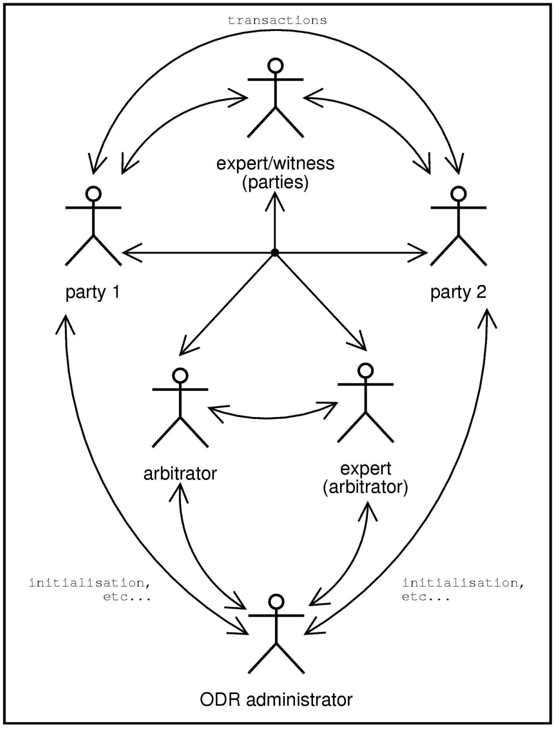




mediation



arbitration



5.3 Scenario of a dispute resolution

5.3.1 Description of scenario

The scenario is composed of two variants. The first one concerns the order of an hardware product, the second one concerns the order of a downloadable software.

Contract conclusion

Variant 1:

The client orders an hardware object through a shop-on-line web site.

Variant 2:

The client orders a software through a shop-on-line web site.

In both cases, the seller uses general conditions, published on the web site, and linked to the order form by something like "by clicking the submit button, you declare that you agree with general conditions (...)".

One of the clauses contained in the general conditions says that if the product sent is defective, the client has the right to get it repaired or replaced, depending on seller's choice.

Litigious contract execution

Variant 1:

The seller sends the hardware. When the client receives it, he notices that it doesn't work.

Variant 2:

The client downloads the software. Then he tries to run it but it doesn't work. He believes that it has not been fully downloaded.

In both variants, the client asks for being refunded and the seller refuses.

Variant1:

The seller accepts that the hardware is defective, but invokes the mentioned clause. He propose to repair the product if the client accepts to pay the transport taxes. The client replies that he will only pay half the taxes.

Variant 2:

The seller denies that the software transmission was faulty. In fact, he believes that the client wants to use his product for free.

In both cases, negotiation in order to friendly resolve the dispute, fails.

ODR process

The client asks an ODR web system for arbitration, submitting his argumentation. He wants to be refunded.

The ODR system accepts to lead the arbitration and notifies the seller. The seller accepts and submits his own argumentation.

Arbitrator actions

The arbitrator's legal decision depends on the fact of a clause, which explicitly excludes the client right to be refunded, being effectively included in the general conditions at the moment of the contract conclusion.

Variant 2:

The decision also depends on the technical proof of the transmitted software default.

The arbitrator asks the seller for contract elements, conditions and, in variant 2, system logs related to the software transmission.

Variant 2:

The technical documents are submitted to an expert.

The arbitrator builds a report containing all documents transmitted, and the expert results. He transmits then the report to both parties.

In both variant, the client contests the presence of a clause excluding his right to be refunded at the moment of the conclusion.

Variant 1:

Complementary documents are sent by the seller, referring to the aborted negotiation about the payment of transport taxes.

The arbitrator transmits to both parties the updated report. And specify a delay of seven days in order to give parties the possibility of commenting it.

Without any comment during this period, the report is considered as having been implicitly accepted.

After the delay period, the arbitrator makes a legal decision and notifies both parties.

Enforcement

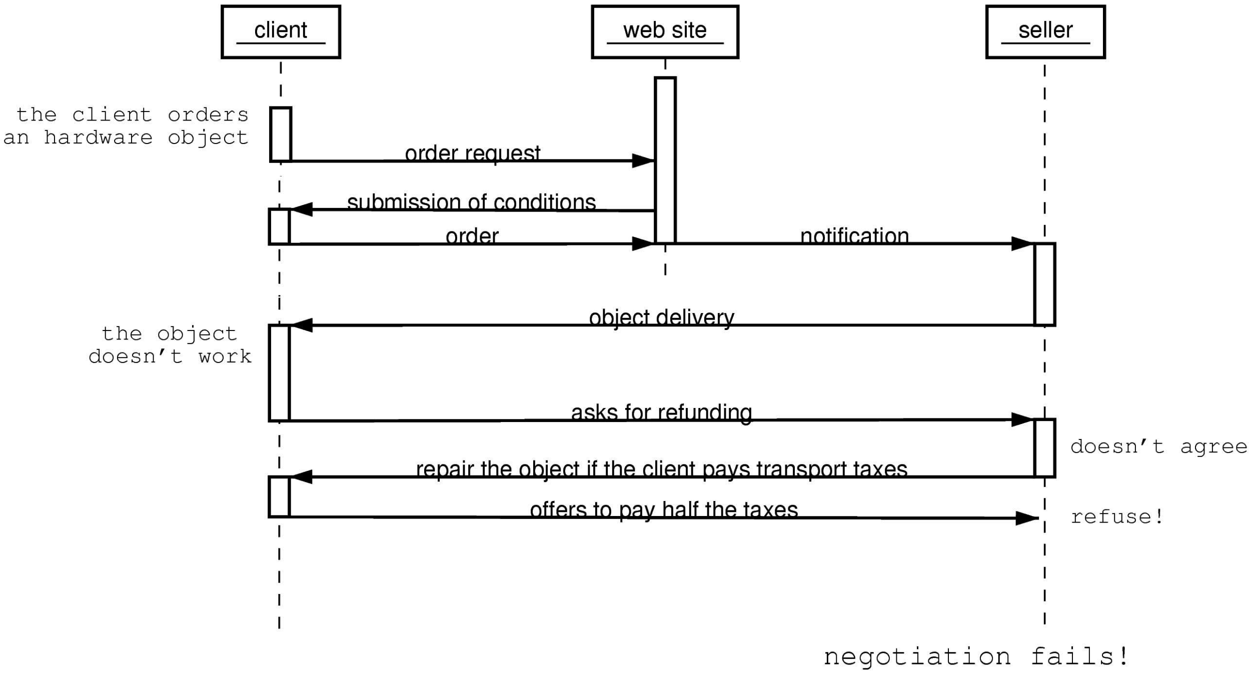
The arbitrator or the ODR administrator validates the decision and publishes it on the ODR web site, accessible to external court potentially asked to apply the decision.

5.3.2 Representation of communication in resolution of a dispute by arbitration

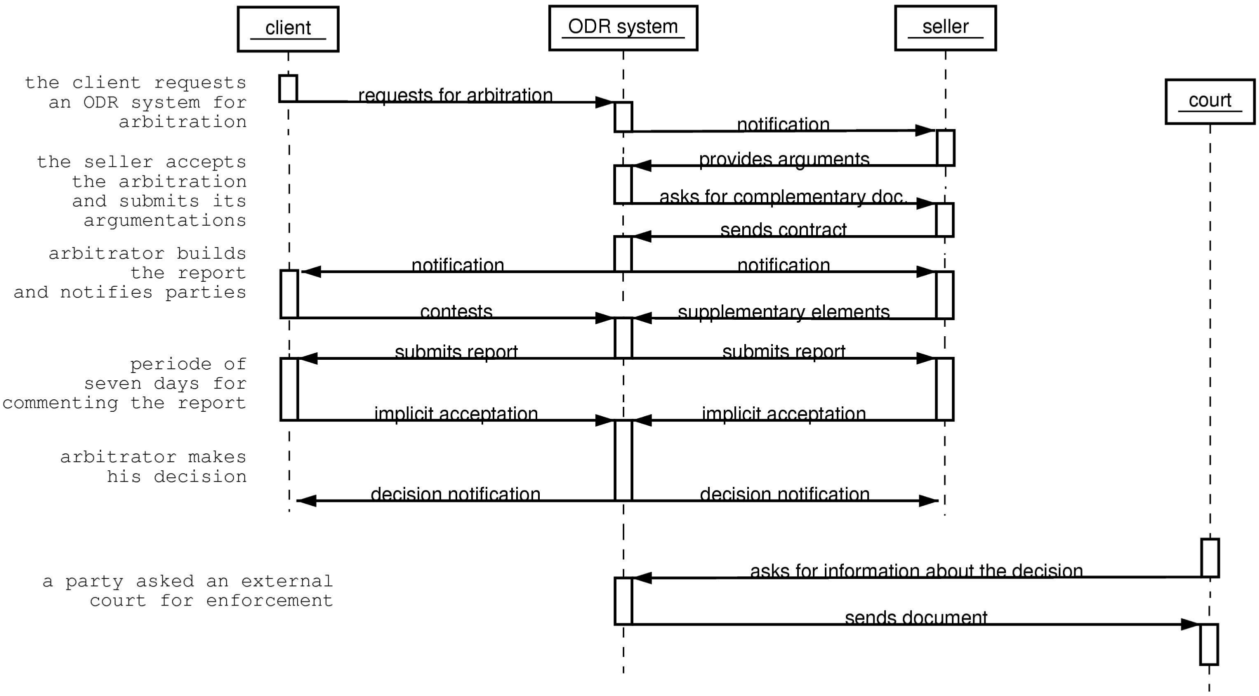
This section describe the different sequence diagram representing the different communications between the different actors implied in the arbitration process.

5.3.2.1 Variant 1

Contract conclusion

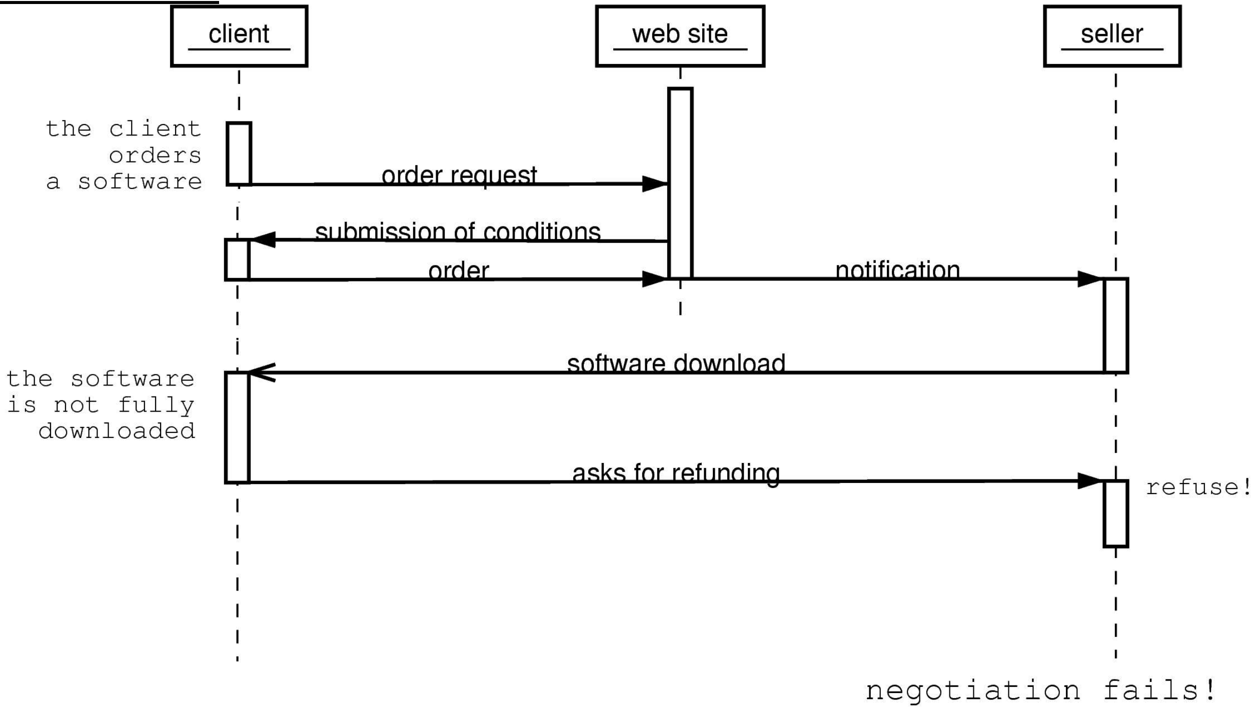


Dispute resolution

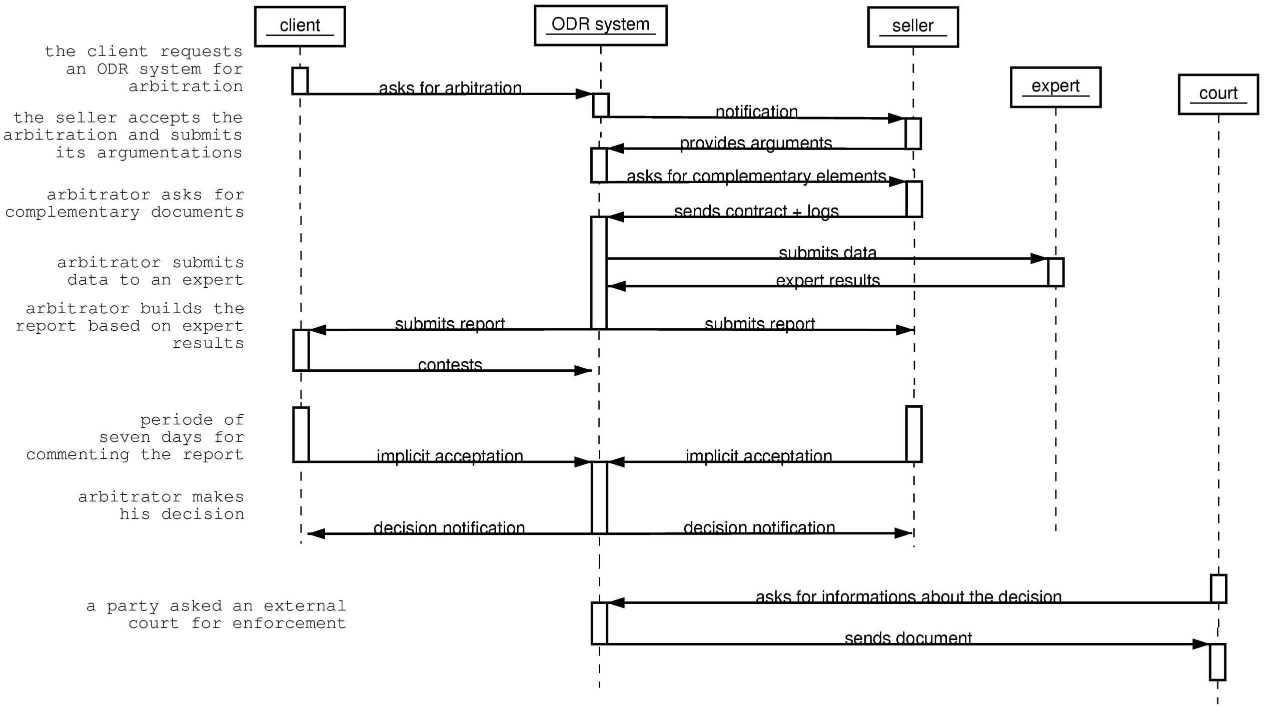

client
ODR system
seller
court
the client requests an ODR system for arbitration
requests for arbitration
notification
the seller accepts the arbitration and submits its argumentations
provides arguments
asks for complementary doc.
sends contract
arbitrator builds the report and notifies parties
notification
notification
contests
supplementary elements
submits report
submits report
periode of seven days for commenting the report
implicit acceptation
implicit acceptation
arbitrator makes his decision
decision notification
decision notification
a party asked an external court for enforcement
asks for information about the decision
sends document

5.3.2.2 Variant 2

Contract conclusion



Dispute resolution


client
ODR system
seller
expert
court
the client requests an ODR system for arbitration
asks for arbitration
the seller accepts the arbitration and submits its argumentations
notification
provides arguments
arbitrator asks for complementary documents
asks for complementary elements
sends contract + logs
arbitrator submits data to an expert
submits data
expert results
arbitrator builds the report based on expert results
submits report
submits report
contests
periode of seven days for commenting the report
implicit acceptation
implicit acceptation
arbitrator makes his decision
decision notification
decision notification
a party asked an external court for enforcement
asks for informations about the decision
sends document

6 Conclusion

Very simply we have seen that ODR Systems must realize their services in interaction with Web-Service Framework. Based on prepared-for-proof exchanged-data and a well-defined interaction model, they could contribute to the enforcement of regulation and marketplace dynamism. Therefore we focus our approach on security needs and specifically non-repudiation property. We are matching them with technical element of Web-Services. We start the modeling of those interactions by scenario case before relaying in further work the discussion on general negotiation protocols [20].

7 Perspectives

To help resolve such conflicts handled by ODR Systems, in business, legal and technological landscapes, many technical issues have to be addressed to increase the quality and reliability of this "online justice". We work on a model in which we integrate ODR Systems as an e-service of normative and regulative class. It implies to resolve dependencies on the data and workflow process in order to treat case reconstruction over standard e-mail, e-communication and electronic marketplace. Preliminarily, this model should prepare transaction to dispute involvement by opening the communication via tracing aspectual e-service. Then we should integrate the e-service as another part, composed of filter (multiplexer) engine and resolver engine in the case of the treatment of a dispute. Finally, we aim to make concrete links between this specific e-service, his management, and the real service offered and enforced by Dispute Resolution Expert Network.

8 References

[1] The BlueBook 2001 - Online Dispute Resolution: The State of the Art and the Issues. T. Schultz, D. Langer, V. Bonnet, G. Kaufmann-Kohler, Juergen Harms. Research report. Dec. 2001.

[2] Electronic Communication Issues related to Online Dispute Resolution Systems
Thomas Schultz, Vincent Bonnet, Dr. Karima Boudaoud, Prof. Gabrielle Kaufmann-Kohler, Prof. Juergen Harms, WWW2002, The Eleventh International World Wide Web Conference, Alternate Paper Tracks, Hawaii, USA, 2002 May 7-11.

[3] Evidence and non-repudiation. Journal of Network and Computer Applications. Zhou and D. Gollmann. London: Academic Press, 1997.

[4] http://www.firstmonday.dk/issues/issue5_8/mccullagh/

[5] Non-repudiation Evidence Generation for CORBA using XML. M. Wichert, D. Ingham, and S. Caughy. In 15th Annual Computer Security Applications Conference, pages 320--327. IEEE Computer Society, Dec. 1999. http://citeseer.nj.nec.com/wichert99nonrepudiation.html

[6] Security In Web Services: An Evolving Threat Model, Shannon Cochran, 2002-05-20.

[7] W3C Recommendation, Extensible Markup Language (XML) 1.0 (Second Edition); see http://www.w3.org/TR/2000/REC-xml-20001006.html

[8] "Simple Object Access Protocol (SOAP) 1.1.", Box, Don, et al. W3C Note. May 2000.
http://www.w3.org/TR/SOAP/ (10 Nov. 2001)

[9] "Web Services Description Language (WSDL) 1.1.", Christensen, Erik, et al. W3C Note. Mar. 2001.
http://www.w3.org/TR/wsdl

[10] UDDI Executive White Paper, uddi.org.
http://www.uddi.org/pubs/UDDI_Executive_White_Paper.PDF
See http://www.uddi.org/ and http://www.uddi.org/faqs.html

[11] Web Services Activity at W3C
http://www.w3.org/2002/ws/

[12] "XML-Signature Syntax and Processing." Bartel, Mark, et al. W3C ProposedRecommendation.Aug.2001.
http://www.w3.org/TR/xmldsig-core/

[13] "XML Key Management Specification (XKMS)." Ford, Warwick, et al. W3C Note. Mar. 2001.
http://www.w3.org/TR/xkms/

[14] OdrXML Draft standard
http://econfidence.jrc.it/default/show.gx?Object.object_id=EC_FORUM0000000000000118C

[15] http://www.w3.org/TR/SOAP-dsig/

[16] SOAP-DSIG and SSL
http://www-106.ibm.com/developerworks/webservices/library/ws-soapsec/

[17] "Oasis Security Services Use Cases and Requirements." Platt, Darren. Oasis SSTC. May 2001.
http://www.oasis-open.org/committees/security/docs/draft-sstc-saml-reqs-01.pdf

[18] "OASIS XACML: Online Application Server Use Cases." Lockhart, Hal. Oasis XAMLC TC. Nov. 2001.
http://www.oasis-open.org/committees/xacml/docs/OnlineServerUseCases.doc

[19] Universal Directories - Web Services as Human Services, Greg FitzPatrick - SkiCal Consortium

[20] A Generic Software Framework for Automated Negotiation
Claudio Bartolini, Chris Preist, Nicholas R. Jennings 1, Trusted E-Services Laboratory, HP Laboratories Bristol, January 23rd, 2002.