Chapter 26

XML-BASED E-CONTRACTING

Michael Merz

Abstract: Under the research projects eLEGAL and OCTANE (funded by the IST programme of the European Commission), European companies have joined forces to develop a variety of tools and templates that are required to support forming and closing electronic contracts over the Internet without any media breaks. This article informs on some of the projects results.

To explore practical business needs, scenarios are taken from different industries: on one side, construction companies and subcontractors enter a joint-venture to accomplish a construction project. In this setting, the e-contracting software is required to let partners compose contracts in an ad-hoc manner out of a clause database. A second scenario shows how bilateral agreements on data exchange relationships, data formats, or rules for information exchange can be made by using e-contracting tools.

Key words: Legal issues, electronic contracting, XML Signature, XML Schema

1. INTRODUCTION

Past decades have witnessed a continuous reduction of cost and media breaks in general office procedures. In many cases, this increase of efficiency has been achieved inside organizations, while inter-organizational processes could only be improved mainly at the very general level of using email. Today, it can be recognized that there is still much room for improvement – specifically in the area of contract editing, negotiation, signing, and fulfillment:

While on one side, processing and exchange of contract data lags behind the formalization of electronic documents used in industrial supply chain applications (orders, call-offs, delivery notes, inventory status, etc.), means for the next step of formalization – and therefore automation – are given

already today (XML, including XML Signatures, certification authorities, collaborative document management systems, etc.).

Research on e-contracting started already in the early 80ies by with a focus on formal contract semantics [1]. Over the 90ies, formalization of business rules and their mark-up started (e.g. [2, 3, 4, 5]) and also security aspects have been explored, e.g. [6, 7]. A first XML-based contract editor was presented by [8] as a result of the COSMOS project (funded under the European ESPRIT programme).

This paper takes a rather practical position and explores business advantages, software applications and future business models that are related to e-contracting.

The presented Contract Editor Software can be downloaded from: http://www.ponton-consulting.de, where it is branded as "Ponton X/E".

2. WHAT IS E-CONTRACTING?

e-Contracting is the process of setting-up, negotiating, signing, and maintaining contracts purely in electronic form.

e-Contracting may either require electronic signatures, or it may take place implicitly by getting a receipt after a retail transaction, it may involve two or many parties and it may consist of a single contract or a set of them – in any case, e-Contracting leads to a legal contractual situation.

E-Contracting affects very different aspects of business life: It is not only an economic act but also a collaboration of two or more parties. It starts with the selection of business partners over the negotiation of positions and modalities and does not even end with the final signing of a document. In many business relations, a contract is not engraved in stone – it is rather the starting point for further refinements, amendments or extensions as the business relation develops its own dynamics.

E-Contracting can thus be viewed from several perspectives: First, a legal viewpoint, second, a business viewpoint and, third, from a document management viewpoint:

1. *Legal Viewpoint.* Here, the legal situation abstracts away from its materialization. A contract might exist in written form, as an oral agreement, or in electronic form – the legal situation is always the same. For lawyers, contracts are abstract concepts that describe this situation and govern the behavior of the parties involved. In practice, many lawyers are occupied with setting-up contract documents that should describe the legal situation as precise as possible.
2. *Business Viewpoint.* In many business environments, a contract is an instrument that is integrated into a long-term relationship: Suppliers and customers agree on contracts for the delivery of a certain quantity of a product at a certain quality, price, and date. These figures will be referred

to and recycled for a long-term period, e.g., in supply-chain relations, supply contracts can be closed for a time span of several years. On the other hand, contract are rather snapshots of the current state of a business agreements. They may be amended, extended, changed or cancelled – all this just requires the agreement of the parties involved and the contract is quickly adjusted to the new situation. From a business viewpoint, the re-use and later change of contracts is key.

3. *Document Management Viewpoint.* Contracts involve more than one party and they are electronic documents, at least over their agreement phase. Therefore, online collaboration is the main facilitator to achieve an agreed and consistent electronic document. Since electronic signing is supported by law today, not only negotiation, but also the joint signing of a contract takes place across the Internet. If the contract is represented in a structured, well-defined form, it may also be interpreted and re-used by other applications such as Enterprise Resource Planning (ERP) systems, Viewers, or archiving tools.

3. THE CONTRACT EDITOR – A TECHNOLOGICAL FRAMEWORK FOR ELECTRONIC CONTRACTS

Business innovation and information & communication technologies (ICT) are often so closely intertwined that it would be a hen-egg loop to determine the actual driver of the innovation process. For the following set of business cases, we will provide as little technology input as necessary. However, to set up the technical environment, at least certain XML-related technologies should be introduced beforehand:

3.1 XML-based legal documents

Many documents in application areas such as supply chain management, electronic publishing, visualisation of database content are based on XML today. However, there is no use of XML in the case of legal documents – only very early standardisation approaches like LegalXML [9] or LexML [10] can be found in the research field. We assume cultural reasons are the main obstacle: Contracts are usually negotiated, exchanged and signed by non-technical persons, therefore tools like word processors and manual exchange by e-mail remain state of the art even today.

On the other hand, contracts are syntactically and semantically highly structured documents. They consist of very specialised clauses, and in highly formalised cases they can even be expressed in terms of a set of parameters – as for example in many form-based standard contracts (car sale, lease, etc.).

For these reasons, it is very likely that contracts will be expressed based on XML in the future. The first approach to achieve this is to define a standard grammar for contractual documents (formally expressed as an XML Schema). This might be a horizontal standard for the representation of contracts in general, or it might be a vertical one for specific applications such as EDI framework contracts, purchase orders, or reinsurance contracts, for example.

3.2 The XML Signature standard

Usually, electronic signatures are created using public key technologies as implemented, e.g., by algorithms such as RSA or DSA [6]. In both cases, the signature is represented by an encrypted hash value derived from a binary document. Usually this is a file of a format like MS Word, PDF, XML, ASCII text, etc. The benefit is that all document types are treated the same way – as a contiguous chunk of bits and bytes. Although this simplifies the process of signing, it complicates signature management at the same time: Signatures and certificates have to be kept as files separate from the original document. If there are several parties signing the same document, the number of managed files consequently increases. In practise, document management systems (DMS) are required in order to maintain document consistency.

While "classic" approaches for electronic signature support signing at the physical level, the XML signature standard allows signing of logical substructures of a document. Thus, an XML document can contain XML structures that are considered relevant for the signature as well as other XML structures that not signed, i.e. they are not within the scope of legal consideration.

Moreover, XML Signature allows embedding of the signature itself into the signed document. This can be done in three different ways: Either the signature becomes a part of the signed document itself (embedded signature). Here, the XML Signature standard requires from technology providers to explicitly exclude any embedded XML Signature sub-elements from the rest of the signed part of a document. The second approach is to embed the signed part of the document into the XML Signature structure. This can be done by copying the signed part from the original document into the signature document (embedding signature). Finally, the signature may be allocated "next to" the signed part, e.g., in a sibling XML path or in a separate document (detached signature).

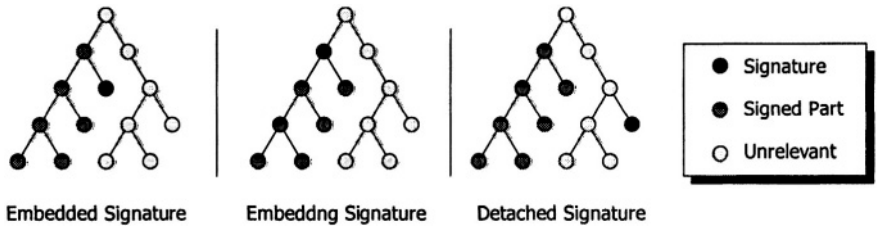

Figure 1: Different relationships of XML Signatures and the signed content within the XML document tree

The signature itself is represented as a standardised XML structure, following the W3C standard (www.w3.org/Signature). Such a <Signature> element not only contains the signature value, but also other data including a reference (URI) to the signed part of the document. In more advanced cases this might even be a list of references that are all signed at once. Also, electronic certificates could be embedded into the XML Signature structure to support signature verification.

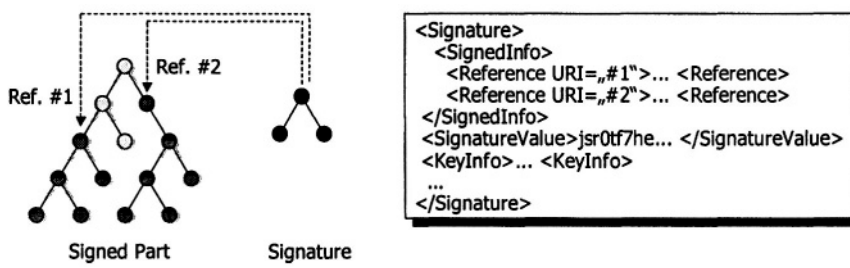

Figure 2: Signing several parts of a document "at one strike"

XML Signature thus helps represent signatures in a much more appropriate form compared to the classic approach. The signing algorithm, however, remains the same: Instead of processing a binary document, XML Signature compliant tools concatenate the value of elements and attributes into a canonical form and apply the signing algorithm (RSA or DSA) to the result. Signature verification is consequently based on the same canonicalized string.

The following figures show how a signed XML document looks like:

Figure 3: An XML document containing an electronic contract with a detached XML Signature

Although this summary might seem a bit technical, we will see further down in the "business process" section how these can be streamlined.

3.3 Contract Management Software

Nobody is willing to understand the binary document format of a word processor. In the same way, nobody is willing to edit an e-contract as an XML document as displayed in figure 3. For this reason, users require a contract editor that facilitates contract processing, specifically creating contract templates, exchanging contract data, filling out contract forms,

signing them and verifying them afterwards. In many business cases e-contracting specifically pays off because contract documents can be directly re-used as input to operational IT systems.

The requirements to contract management systems are therefore:

- *User interface:* Conceal syntactic XML details from the user.
- Support users or consultants in *defining structure and layout* of a contract.
- Support the users in easily *composing contract clauses.*
- In case of *standard contracts,* support users in entering parameters for the clauses.
- Assure syntactic and semantic correctness of the contract as far as possible.
- Support the signing and signature verification process.
- Allow for additional processing options like transformation into other data formats, integration of application-level functions (e.g. selection from an online catalogue), and access to a document management system that provides functions for document storage, user rights management, and archiving of documents.

3.4 Contributions from the OCTANE and eLEGAL projects

As a result of two projects – eLEGAL and OCTANE – funded by the IST programme of the European Commission, a contract editor software was developed.

The Contract Editor can be divided into two different parts: First, the *base software* can be used to edit any XML document, with optional support of a customisable document layout. The base software provides an interface to *plug-in* additional functions, such as an electronic signing tool, an XSL-based HTML converter, remote access to a document management system, and e-mail support to send the XML document directly to a business partner. This part of the software was developed for the *OCTANE* project.

One of the most complex plug-ins is the *Contract Wizard.* This tool requires a certain set of well-known XML substructures in the contract consisting of nested <Clause> elements. The purpose of the Contract Wizard is let the user select clause types and clauses from a database and insert then into the contract. Moreover, clauses can be checked for consistency using rule definitions. This part was developed for the *eLEGAL* project.

3.4.1 The Contract Editor

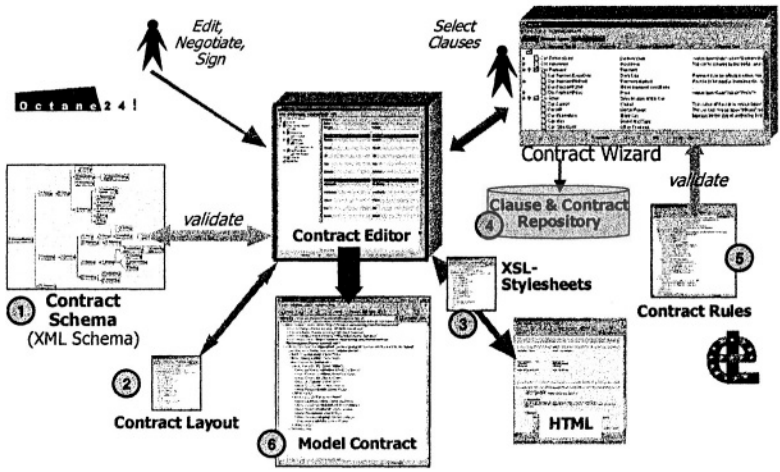

Figure 4: Components of the Contract Editor

Figure 4 illustrates the different document types and data sources used in creating an electronic contract:

1. An *XML Schema* is used to define the structure of the contract. Based on this specification, the Contract Editor does not allow any document structure to be created which does not comply with the underlying XML Schema.
2. By default, no *contract layout* is needed. In this case, all data elements are displayed in canonical order and in a vertical flow.
3. Another option is to develop *mapping definitions* for converters. Under OCTANE, a HTML converter was developed based on XSL transformations. This may be extended by MS Word converters or other transformations in the future.
4. The *Contract Wizard* uses a database to retrieve clause types and clause texts. The data model for clauses and clause types is illustrated further down.
5. To structure database access, *contract rules* can be defined by the creator of a contract template. These rules define which clause types are to be used for a specific contract type, whether the clauses are mandatory or optional, and which logical constraints apply to them.
6. Finally, model contract can be created based on the given tools by legal experts. They can be published or used as templates for actual contracts.

In short, the Contract Editor deals with syntactic aspects of XML-based contracts, while the Contract Wizard deals with semantic aspects of the content.

The following two figures illustrate the use of the Contract Editor for a reporting task between printers and their customers (Inventory Stock Report). The first figure displays the document without a layout definition, whereas the second illustrates the application of a layout definition:

The screenshot in figure 5 shows the default user interface to edit a Schema-based XML document. A layout definition is applied in figure 6. Here, the user interface is structured much better, and the data elements are displayed with German labels.

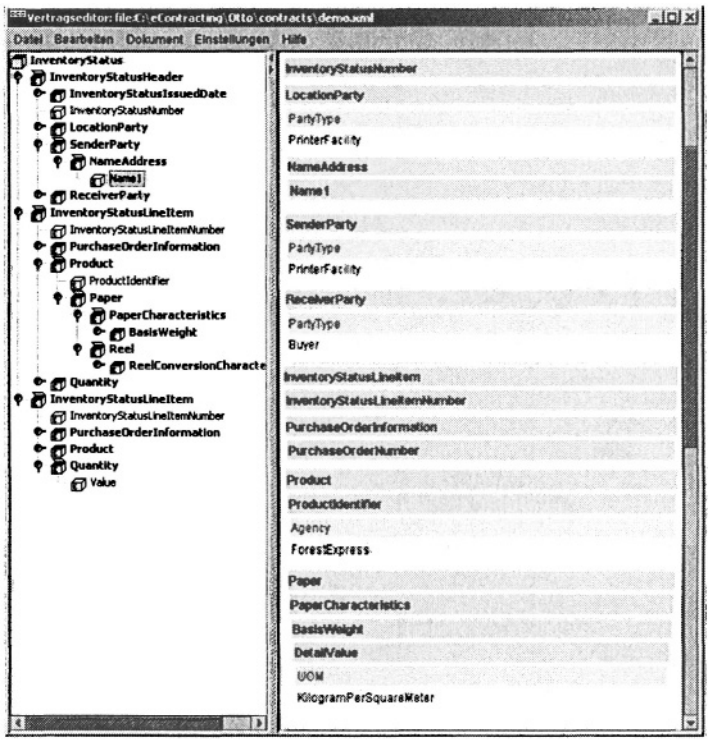

Figure 5: Inventory Status report displayed in canonical form

One of the most interesting features of the Contract Editor is the signing of contracts based on the XML Signature standard. In principle, the Contract Editor allows you to sign any part of an XML document and place the signature in any other part. To define these locations, several settings have to be provided at the level of the contract template:

- Which part of the document is to be signed?
- Which part will carry the signature?

- Will the contract be signed by more than one party? If so,
  - Which element contains the list of parties?
  - Which element represents a party?
  - Which element contains the name of the party, as it should be displayed so that the user can choose the party to sign for?

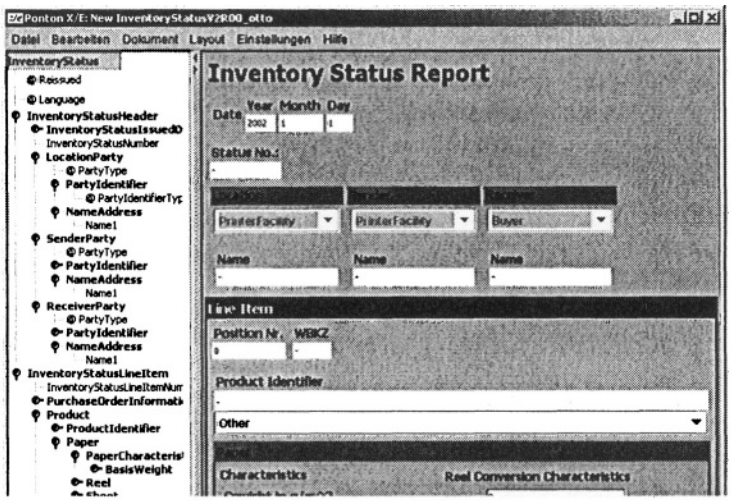

Figure 6: Inventory Status report applying a layout definition with German localisation

The signing plug-in of the software expects some well-known XML attributes defined within the document. These attributes indicate the part to be signed, where the signature(s) will be stored, and where the party descriptions are located.

This way, any XML document can be extended with minimal effort to accept XML Signatures.

3.4.2 Using the Contract Wizard

The Contract Wizard is a plug-in that allows contract clauses to be composed for a given legal context. This context is defined by the language, the applied legislation and the domain (e.g. a purchasing contract, a lease, a consortium agreement, or company statutes, just to name a few).

To use the Contract Wizard, one precondition must be met by the XML Schema for the underlying e-contract: It has to support a nested structure of text clauses, as in the following example:

```
<Clauses>
    <Clause ClauseID="XXX" ClauseTypeID="XXX">
        <ClauseTitle>XXX</ClauseTitle>
        <ClauseText>abcde pqr stuvwx yz</ClauseText>
```

```
      </Clause>
      <Clause ClauseID=" XXX" ClauseTypeID="XXX">
            <ClauseTitle>XXX</ClauseTitle>
            <ClauseText>abcde pqr stuvwx yz</ClauseText>
            <Clause ClauseID=" XXX" ClauseTypeID="XXX">
                  <ClauseTitle>XXX</ClauseTitle>
                  <ClauseText>  abcde  fghij  klmno  pqr …
                  </ClauseText>
            </Clause>
      </Clause>
      …
</Clauses>
```

This is the target structure the Contract Wizard operates on. figure 7 shows an End User License Agreement (EULA) as another example. Here, a set of clauses has already been composed giving a deeply structured tree of <Clause> elements. This can be seen on the left hand side in the layout tree and in the content window showing the top-level clause "Restriction of Use" with subclauses in the subordinated window.

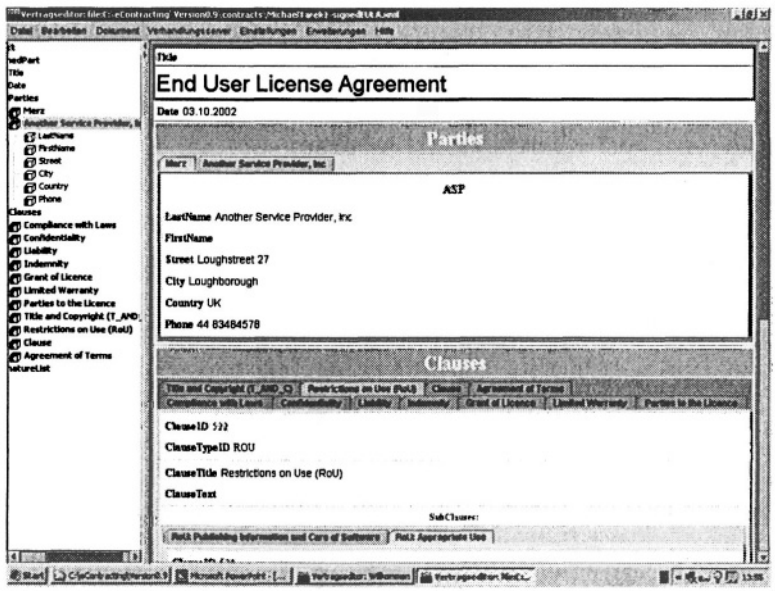

Figure 7: An electronic contract with a nested structure of <Clause> elements

4. CONCLUSION

In this paper some of the concepts and the developments performed within the framework of both eLEGAL and OCTANE Projects have been presented. By means of the eContracting scheme and tools developed within the eLEGAL Project, such development lead to an environment able to establish a sound and legally valid business relationship fostering a new business model for B2B scenarios and consultants seeking to broaden their business activities capturing the opportunities provided by the available ICT tools and infrastructures.

REFERENCES

1. Lee, R.M – A Logical Calculus for Describing Financial Contracts, PhD Dissertation, Department of Decision Sciences, The Wharton School, University of Pennsylvania, 1980
2. Zoran Milosevic, David Arnold, Luke O'Connor, Inter-enterprise Contract Architecture For Open Distributed Systems: Security Requirements, WET ICE'96 Workshop on Enterprise Security, Stanford, USA, June 1996.
3. Michael Schroeder, Gerd Wagner, eds.: Proceedings of the International Workshop on Rule Markup Languages for Business Rules on the Semantic Web, 14 June 2002, Sardinia, Italy, CEUR Workshops, http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/
4. Rule Markup Initiative Web site, http://www.dfki.uni-kl.de/ruleml/
5. Laurence L. Leff: Automated Reasoning with Legal XML Documents; *Conference '00,* Month 1-2, 2000, City, State.
6. Bruce Shneier: Applied Cryptography, Second Edition, John Wiley & Sons, 1996
7. Michael Gisler, Harald Häuschen, Yvonne Jöhri, Arion Meier, Otto Müller, Bernd Schopp, Katarina Stanoevska: Requirements on Secure Electronic Contracts, Report No. mcm-institute-1999-01, Institute for Media and Communication Management, University St. Gallen, 1999
8. M. Merz, F. Griffel et al, Supporting Electronic Commerce Transactions with Contracting Services, International Journal of Co-operative Information Systems, 1998, p. 16.
9. Web site of the Legal XML standardisation group: www.legalxml.org
10. Web site of the Lex ML research group: http://www.lexml.de/mission_english.htm